# Preprocessing to Deal with Hard Problems

Upper and Lower Bounds for Kernelization of Graph Problems

DISSERTATION

zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftliche Fakultät
Humboldt-Universität zu Berlin

von
**M.Sc. Eva-Maria Christiana Hols**

Präsidentin der Humboldt-Universität zu Berlin:
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftliche Fakultät:
Prof. Dr. Elmar Kulke

Gutachter:
1. Prof. Dr. Stefan Kratsch, Humboldt-Universität zu Berlin
2. Prof. Dr. Johannes Köbler, Humboldt-Universität zu Berlin
3. Prof. Dr. Rolf Niedermeier, Technische Universität Berlin

**Tag der mündlichen Prüfung:** 25. Februar 2020

## Abstract

In classical complexity theory we distinguish between the class $\mathsf{P}$, of polynomial-time solvable problems, and the class $\mathsf{NP}$-hard, of problems where the widely-held belief is that we cannot solve these problems in polynomial time. Unfortunately, many of the problems we want to solve are $\mathsf{NP}$-hard and we still need to solve them as fast as possible. At the same time, there is a large discrepancy between the empirically observed running times and the established worst-case bounds. Using preprocessing or data reductions on real-world instances is known to lead to huge improvements in the running time. Here we come to the limits of classical complexity theory because it lacks suitable models for studying efficient preprocessing.

In this thesis, we focus on preprocessing algorithms for $\mathsf{NP}$-hard problems, i.e., algorithms that compute in polynomial time an equivalent instance of smaller size or with certain properties. Moreover, we prove theorems about how successful such algorithms are. Since we consider $\mathsf{NP}$-hard problems, we cannot expect that we can reduce the size of every input instance; otherwise, we would show that $\mathsf{P} = \mathsf{NP}$. Our goal is to find ways to preprocess at least certain instances of an $\mathsf{NP}$-hard problem by considering the structure of the input instance. More precisely, given an instance and an additional parameter $\ell$, we want to compute in polynomial time an equivalent instance whose size and parameter value is bounded by a function in the parameter $\ell$ only. Such preprocessing algorithms are called *kernelization algorithms* in the field of parameterized complexity.

We will consider three $\mathsf{NP}$-hard graph problems, namely Vertex Cover, Edge Dominating Set, and Subset FVS. For Vertex Cover we will unify known results for kernelization algorithms when parameterized by the size of a deletion set to a specified graph class $\mathcal{C}$. We point out, among other things, that bounded minimal blocking set size in $\mathcal{C}$ is necessary but not sufficient to obtain a polynomial kernelization. Then, we determine upper and lower bounds for the largest minimal blocking set size for some graph classes and use the upper bounds to derive polynomial kernelizations for Vertex Cover when parameterized by the size of a deletion distance to such a graph class. Afterwards, we study the existence of polynomial kernelizations for Edge Dominating Set when parameterized by the size of a deletion set to a graph class $\mathcal{C}$. Here, we first show that Edge Dominating Set parameterized by the size of a deletion set to a disjoint union of paths of length two has no polynomial kernelization (under standard assumptions) which points out that the existence of polynomial kernelizations is much more complicated than for Vertex Cover. In a second step, we fully classify for all finite sets $\mathcal{H}$ of graphs, whether there exists a polynomial kernelization, when the parameter is the size of a deletion set to a disjoint union of graphs that are contained in $\mathcal{H}$. Finally, we consider graph cut problems, especially the Subset FVS problem. We show in two steps that Subset FVS has a randomized polynomial kernelization when the parameter is the solution size, using tools from matroid theory.

## Zusammenfassung

In der klassischen Komplexitätstheorie unterscheiden wir zwischen der Klasse P von Problemen die in Polynomialzeit lösbar sind, und der Klasse NP-schwer von Problemen bei denen die allgemeine Annahme ist, dass wir diese nicht in Polynomialzeit lösen können. Unglücklicherweise sind viele der Probleme, die wir lösen möchten, NP-schwer, trotzdem müssen wir diese Probleme so schnell wie möglich lösen. Gleichzeitig besteht eine große Diskrepanz zwischen den empirisch beobachteten Laufzeiten und den festgestellten worst-case Laufzeiten. Es ist bekannt, dass die Verwendung von Vorverarbeitung oder Datenreduktion auf realen Instanzen zu enormen Laufzeitverbesserungen führt. Hier stoßen wir an die Grenze der klassischen Komplexitätstheorie, da geeignete Modelle für die Studie effizienter Vorverarbeitung fehlt.

Der Fokus dieser Arbeit liegt auf Vorverarbeitungsalgorithmen für NP-schwere Probleme, d.h. Algorithmen, die in Polynomialzeit eine äquivalente Instanz bestimmen, die eine kleinere Größe oder bestimmte Eigenschaften hat. Darüber hinaus beweisen wir Theoreme darüber, wie erfolgreich solche Algorithmen sind. Da wir NP-schwere Probleme betrachten, können wir nicht erwarten, dass wir die Größe jeder Instanz reduzieren können. Andernfalls würden wir zeigen, dass P = NP gilt. Unser Ziel ist es, Wege zu finden, um zumindest bestimmte Instanzen eines NP-schweren Problems vorverarbeiten zu können, indem wir die Struktur der Instanz betrachten. Genauer gesagt, für eine gegebene Instanz und einen zusätzlichen Parameter $\ell$, möchten wir in Polynomialzeit eine äquivalente Instanz berechnen, deren Größe und Parameterwert nur durch eine Funktion im Parameterwert $\ell$ beschränkt ist. Solche Vorverarbeitungsalgorithmen werden im Bereich der parametrisierten Komplexitätstheorie *Kernelisierung* genannt.

Wir werden drei NP-schwere Graphenprobleme betrachten, nämlich Vertex Cover, Edge Dominating Set und Subset FVS. Für Vertex Cover werden wir bekannte Ergebnisse für Kernelisierungen vereinheitlichen, wenn der Parameter die Größe einer Knotenmenge ist, deren Entfernung in einem Graphen einer gegebenen Graphklasse $\mathcal{C}$ resultiert. Wir zeigen unter anderem auf, dass eine beschränkte Größe der minimalen blocking sets in $\mathcal{C}$ erforderlich ist, jedoch nicht ausreicht, um eine polynomielle Kernelisierung zu erhalten. Danach bestimmen wir obere und untere Schranken für die Größe des größtmöglichen minimalen blocking sets für einige Graphklassen und wir verwenden die oberen Schranken, um polynomielle Kernelisierungen für Vertex Cover zu erhalten, wenn der Parameter die Größe einer Knotenmenge ist, deren Entfernung in einem Graphen einer solchen Klasse resultiert. Anschließend untersuchen wir die Kernelisierbarkeit von Edge Dominating Set, wenn der Parameter die Größe einer Knotenmenge ist, dessen Entfernung in einem Graphen einer gegebenen Graphklasse resultiert. Hier zeigen wir zuerst, dass Edge Dominating Set parametrisiert durch die Größe einer Knotenmenge, deren Entfernung in einem Graphen resultiert, der eine disjunkte Vereinigung von Pfaden der Länge zwei ist, keine polynomielle Kernelisierung hat (unter Standardannahmen). Dies weist darauf hin, dass die Kernelisierbarkeit deutlich komplexer ist als für Vertex Cover. In einem zweiten Schritt klassifizieren wir für alle endlichen Mengen $\mathcal{H}$ von Graphen, ob es eine polynomielle Kernelisierung gibt, wenn der Parameter die Größe einer Knotenmenge ist, deren Entferung in einem Graphen

resultiert, der eine disjunkte Vereinigung von Graphen der Menge $\mathcal{H}$ ist. Schließlich betrachten wir Graph-Cut Probleme, insbesondere das Subset FVS Problem. Wir zeigen in zwei Schritten, dass Subset FVS eine randomisierte polynomielle Kernelisierung hat, wenn der Parameter die Lösungsgröße ist, und benutzen hierbei Werkzeuge aus der Matroid-Theorie.

## Acknowledgement

First of all I want to thank my supervisor Stefan Kratsch for his continuous support. He introduced me to the field of parameterized complexity and shared his deep insights into the field. Whenever I had a question he took the time to discuss them with me and also asked the right questions to guide my work. You always knew how to motivate and encourage me. Thank you for all your advice regarding research, writing and academic environment.

Furthermore, I want to thank Astrid Pieterse for co-authoring. I really enjoyed working with you. In addition I would like to thank my colleagues, Falko Hegerfeld and Florian Nelles for reading parts of this thesis and providing me with helpful comments. It was a pleasure to work with you. I also want to thank Anneke Buffone, Alex Buffone and Martin Lambertz for helping me to improve the introduction and the conclusion.

The Deutsche Forschungsgemeinschaft funded my research during the time in Berlin within the Emmy Noether-grant (KR 4286/1), for which I am very thankful. Next, I would like to thank in advance my dissertation committee for taking the time to read this thesis.

I also want to thank two of my former teachers, Erich Wilberg and Wolfgang Miletzki. They taught me to believe in myself and not to be afraid of asking questions. A special thanks to my family as well as my future family for their endless support and telling me to make breaks here and there.

Last but foremost, I want to thank Timm for always supporting me, being my skull, dealing with me being grumpy when something did not went the way I wanted it to, and reading every paper as well as this thesis. I could not have done this without you.

# CONTENTS

# Part I.

# Foundations

# CHAPTER 1 INTRODUCTION

As humans we do preprocessing and data reduction all the time. Whenever we have to make a decision, for example choosing a museum we want to visit on Monday, we ignore solutions that fail or that are not possible. In our example, we can ignore every museum that is closed on Monday or in a different city (i.e. not reachable in one day). Furthermore, we concentrate on the parts of the problem that affect us. For our example, the weather is (in general) irrelevant for our decision which museum we choose. Thus, we are able to simplify the problem to its *core*.

Unsurprisingly, computer programs preprocess data all the time and this approach yields good results in practice. A good example of this is the *simplex algorithm* which is used to solve linear programs. Intuitively, a linear program consists of a set of constraints, for example, limited capacity of machines, and a target function where the goal is to minimize or maximize the value of the target function, for example, maximizing the profit or minimizing the cost. The simplex algorithm applies some advanced preprocessing rules to reduce the number of constraints. Furthermore, it reduces the solution space to the extremal points (the vertices of the polytope described by the constraints).

This raises the questions whether preprocessing is always possible or whether there is a limit on how much we can preprocess a given problem. In general, the answer to both questions is negative because a positive answer to one of these questions would answer the Millennium Prize Problem $\mathsf{P}$ versus $\mathsf{NP}$ positively whereas it is widely believed that $\mathsf{P} \neq \mathsf{NP}$. Simply put, a problem belongs to the class $\mathsf{P}$ when a computer can solve the problem efficiently (in relation to the input size), whereas the class $\mathsf{NP}$ contains all problems where a computer can verify a given solution to the problem efficiently (in relation to the input size), but there may not be an efficient way to find a solution. Thus, it holds that $\mathsf{P} \subseteq \mathsf{NP}$.[1] The question whether $\mathsf{P} = \mathsf{NP}$ led to the introduction

[1] One can reformulate the question whether $\mathsf{P} = \mathsf{NP}$ to the question whether every problem where we can verify a solution efficiently can also be solved efficiently.

of the concept of NP-completeness by Cook [Coo71]. Intuitively, a problem is NP-complete if it is contained in NP and if an efficient (i.e. polynomial time) algorithm for this problem implies efficient algorithms for all problems in NP. Therefore, we believe that NP-complete problems are hard to solve. Consequently, a preprocessing algorithm that reduces any instance of an NP-complete problem efficiently to a strictly smaller instance would imply that $\mathsf{P} = \mathsf{NP}$ because we shrink the instance in each step until it is solved or the size is so small that guessing the solution is efficient.

The problem with NP-completeness is that it only tells us that there are instances of a problem which are hard to solve, and it does not mean that every problem instance is hard to solve. A good example for this are combinatorial puzzles, like Sudoku, Lights Up, or Minesweeper. All of these puzzles are NP-complete [KPS08], but people solve them all the time using preprocessing and not by trying out all possible solutions.

Let us consider the following problem to get an intuition about what makes instances of certain problems hard, and how far we can preprocess certain instances. To this end, suppose an intelligence agency wants to monitor all traffic on the internet. They can achieve this by positioning agents at internet exchange points (IXP) such that every cable can be monitored by at least one agent, i.e., an agent is positioned at at least one of the two endpoints of every cable.[2] For our example we choose the following cities

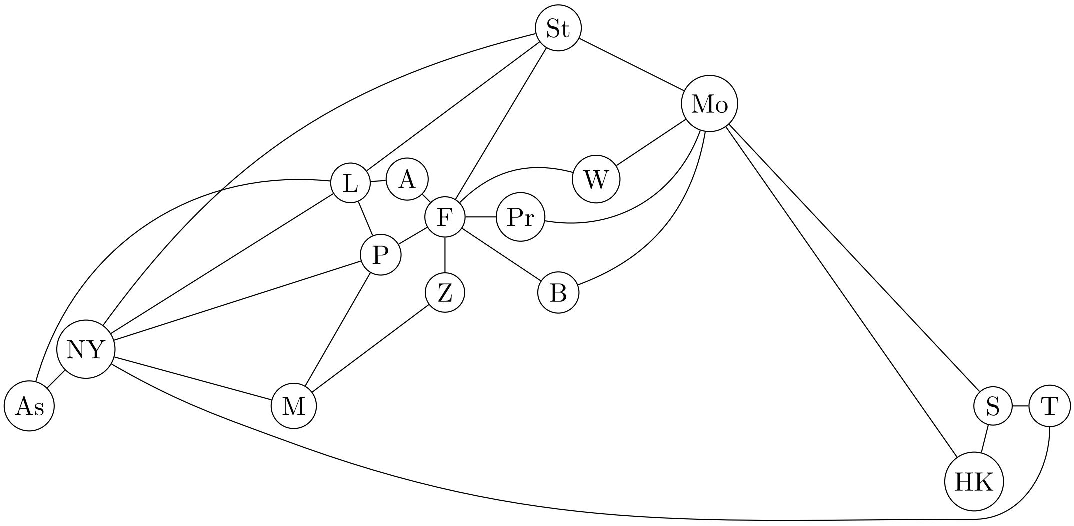



Figure 1.1.: Example: Network of internet exchange points.

with IXPs: Frankfurt, Amsterdam, London, Moscow, Zürich, Seoul, Tokyo, Stockholm, Madrid, Ashburn (USA), Budapest, Warsaw, Hong Kong, New York, Paris and Prague. The cables between these IXPs are depicted in Figure 1.1.

We have 16 IXPs in our example. Obviously, we could position one agent at each IXP, but this might be too expensive. Thus, we want to find the smallest number

[2]This problem corresponds to the well-known VERTEX COVER problem.

of agents to monitor all cables. Naively, we could try all possible sets of positions of agents which leads to $2^{16} = 65536$ possible arrangements of the agents from which we can choose an optimal one. But perhaps we have at most six agents and want to know whether this is possible. What can we do? Of course, we could try all possibilities to position these six agents which still yields to $\binom{16}{6} = 8008$ possible arrangements of the agents. However, if there exists an IXP that is connected via cable to at least seven IXPs, then one agent must be positioned at this point. Otherwise, we have to position an agent at each of the at least seven IXPs. But this exceeds the number of available agents.[3]

Considering our example, we observe that the IXP in Frankfurt (F) is connected via cable to seven other IXPs. This allows us to preprocess our instance because we know that an agent must be positioned here under the condition that there exists a solution with only six agents. From now on we can ignore the cables that have one endpoint in Frankfurt. Now, we have to monitor the remaining cables with only five agents. Since the IXP in New York City (NY) is connected to six other IXPs, again, we know that we have to position an agent at this IXP. Thus, we can ignore the cables that have one endpoint in Frankfurt or New York City, and have four agents to monitor the remaining cables. By continuing this process we can also figure out that we have to position an agent in Moscow (Mo) and London (L). This leaves us with only two agents that we have to position on the remaining 12 IXPs which reduces the number of possibilities significantly.

It is well known that the problem of positioning agents to monitor all internet traffic with a given number of agents is NP-complete. But, as seen in our example, there are instances where we can use preprocessing to solve the problem considerably more efficiently. For our instance, after the preprocessing, one can easily see that the two remaining agents have to be positioned in Madrid (M) and Seoul (S).

As mentioned before, this is the crux of the classification of NP-completeness: It only tells us that there exist some instances where we have to consider nearly every possible solution to find out whether there exists a feasible solution and where preprocessing fails. It does not mean that every instance of the problem is hard.

Combinatorial puzzles, like Sudoku, are designed such that we can solve them using preprocessing (combinatorics). But why does preprocessing work so well on real-world data? Many real-world instances have some properties that allow us to preprocess a given instance. When developing algorithms we use preprocessing rules (heuristics) to take advantage of these properties. Yet, we do not know how far such heuristics reduce a given instance and their benefit is only shown empirically. This raises the question what makes some instances so hard compared to "easier" instances of the problem?

Here, we come to the limit of classical complexity theory. The simple classification of easy, class P, and hard, class NP-complete, is not enough to answer the question why preprocessing works on many instances of NP-complete problems. We need to consider the structure of instances to explain why some instances of NP-complete problems are

[3] This is a well-known reduction rule for Vertex Cover.

easy to solve. To capture certain difficult structures one started to use a multivariate approach to obtain a better understanding of hard problems and to understand which structures cause the hardness.

In a multivariate approach an instance of a given problem is associated with a secondary measurement. This measurement could, for example, be the size of the solution we want to find. Now, instead of expressing the complexity of a problem by the overall input size only, the secondary measurement is taken into account. Thus, we express the running time, that we need to solve the problem, by the overall input size as well as by the secondary measurement, which yields a more detailed classification of hardness. This multivariate approach is known as *Parameterized Complexity* and was pioneered in the early nineties by Downey and Fellows [DF92b, DF99]. Simply put, we want to figure out the structures that cause the hardness of a given NP-complete problem, and then we want to find out whether the problem becomes easy when this structure has a limited size.

What are the structures that make a problem hard to solve? Can we measure these problematic structures? Can we preprocess a given instance to a size that only depends on the measure of these problematic structures? In this thesis, we consider several graph problems and we show that although all of these graph problems are hard to solve in general (NP-complete) the above questions lead to very different answers.

**Parameterized Complexity.** As mentioned before, parameterized complexity classifies problems on a more detailed level. In contrast to classical complexity theory, where the running time depends only on the size of the input instance of a given problem, parameterized complexity analyzes the running time in terms of the input size as well as another value, called the *parameter*, to capture the difficulty of the given problem. In the example above, this extra parameter could be the number of available agents. We say that a problem is *parameterized* when each instance of the problem is related to a fixed parameter value.

The goal of parameterized complexity is to design algorithms that are efficient when the parameter is small (constant). Consequently, we want to work out what makes a problem hard to solve. Afterwards, we want to find a measurement that describes the extent of the property that makes a problem hard to solve. Using this understanding, we design algorithms that are efficient when this measure is small, which means that the NP-complete problem can be solved efficiently when this measure is small. This also captures the fact that many NP-complete graph problems are easy on forests[4]. A possible parameter could be a measurement which describes how far a given graph is from being a forest, and this measure is a fixed constant when the graph is a forest.

How can such running times look like? The running time can, for example, be a polynomial in the input size $n$ where the exponent depends on the parameter $\ell$, i.e., $n^{f(\ell)}$ where $f$ is a computable function, or a polynomial in the input size $n$ multiplied by a function depending on the parameter $\ell$, i.e., $f(\ell) \cdot n^c$ for a constant $c$ and a

[4]A forest is a graph that contains no cycle.

computable function $f$. Obviously, both running times are efficient when $\ell$ is constant. Nevertheless, a running time of $f(\ell)\cdot n^c$ is preferable over a running time of $n^{f(\ell)}$ because the polynomial is independent of the parameter. For large instances a polynomial with an exponent of 16, for example, is impractical whereas multiplying a polynomial of hopefully small degree $c$ by even a large constant $f(\ell)$ has more success to run reasonably efficient in practice. We call algorithms that have a running time of $f(\ell) \cdot n^c$ *fixed-parameter tractable* algorithms, or short, *fpt*-algorithms. By FPT we denote the class of all problems that are *fixed-parameter tractable*, i.e., parameterized problems for which an fpt-algorithm exists. The concept of fpt-algorithms plays a central role in parameterized complexity. Parameterized problems, which can be solved by an algorithm where the running time is of the form $f(\ell) \cdot n^{g(\ell)}$, belong to the class XP. Obviously, it holds that FPT $\subseteq$ XP.

Let us consider two NP-complete graph problems that are equivalent in classical complexity theory but behave differently under parameterized complexity, namely Vertex Cover and Independent Set[5]. A set of vertices $X$ of a given graph $G = (V, E)$ is a vertex cover if and only if the complement $I = V \setminus X$ is an independent set of $G$. Thus, an algorithm for one of these problems yields an algorithm for the other problem. However, when we look at the parameterized version of these problems, where the parameter $\ell$ is the solution size, they behave differently. Obviously, both can be solved in $|V|^{\mathcal{O}(\ell)}$ time by considering all subsets of vertices of size $\ell$. For Independent Set there is no fpt-algorithm known, even worse it is well known that Independent Set is $W[1]$-complete [DF95] which means that we do not hope to find an fpt-algorithm since the common assumption is that FPT $\neq W[1]$. On the contrary, Vertex Cover belongs to the class FPT and the best known algorithm runs in time $\mathcal{O}(1.2738^\ell + \ell|V|)$ [CKX10].

**Kernelization.** So far, we have seen that it is unlikely that we can reduce every instance of an NP-complete problem efficiently (unless P = NP). This is mainly attributable to the fact that we do not believe that we can solve any NP-complete problem efficiently. However, the framework of parameterized complexity, or more precisely, the notation of fixed-parameter tractability allows us to tell when a problem is solvable efficiently depending on a given parameter. Therefore, the following two questions arise: Can we use the framework of parameterized complexity to define efficient preprocessing? And, can we get a decent guarantee for the preprocessing when a problem belongs to the class FPT?

The answer to the first question is yes. Using parameterized complexity we are able to measure the efficiency of preprocessing rules. More precisely, depending on the parameter $\ell$ we are able to bound the size of the reduced instance or we can tell that there exists no guarantee, meaning that for each function $f$ depending on $\ell$ there are instances that we cannot reduce to size $f(\ell)$. More formally, an algorithm that reduces a given parameterized problem with parameter $\ell$ in polynomial time to an

[5]Given a graph $G$, a set $X$ or $I$ of vertices is a vertex cover or an independent set of $G$, respectively, if every edge of $G$ has at least one endpoint in $X$ or at most one endpoint in $I$, respectively

equivalent instance of the same parameterized problem whose size (and parameter value) only depends on the parameter value $\ell$ is called a *kernelization algorithm*, or short *kernelization*. The reduced instance is called a *kernel*.

The function $f$ that bounds the size of the kernel can depend exponentially on the parameter $\ell$. But, in general, we are interested in so-called *polynomial kernelizations* which means that the function $f$ is a polynomial. Observe, since we reduce a given instance to size $f(\ell)$ we cannot hope to solve a given instance by only using reduction rules. However, we reduce the instance to the *difficult core*.

Now, let us return to our second question, whether parameterized problems that are fixed-parameter tractable with respect to a parameter $\ell$ always have a kernel. The answer is yes, when we ask for any kernel, and likely no, when we ask for a polynomial kernel. Obviously, having a kernel for a (decidable) parameterized problem implies that the problem is also fixed-parameter tractable: We first apply the kernelization algorithm to a given instance and subsequently apply any brute-force algorithm to the kernelized instance (whose size depends only on the parameter $\ell$) to obtain an fpt-algorithm. The other direction, that an fpt-algorithm implies a kernel, also holds (see for example [Bod09]): Suppose that a parameterized problem admits an fpt-algorithm which runs in time $f(\ell) \cdot n^c$. We run the fpt-algorithm for $n^{c+1}$ many steps. If the algorithm terminates in this time then we return a trivial yes- or no-instance depending on the output. Otherwise, we know that the algorithm does not terminate after $n^{c+1}$ steps, but the algorithm terminates after $f(\ell) \cdot n^c$ steps which implies that $n \leq f(\ell)$. Thus, we return the instance itself whose size is bounded by $f(\ell)$. Overall, we showed that a (parameterized) problem belongs to the class FPT if and only if it is decidable and admits a kernel.

In the following, we consider two graph problems that are NP-complete and belong to the class FPT. We will give some intuition why one of these problems admits a polynomial kernel whereas it is unlikely that the other problem has a polynomial kernel.

Vertex Cover parameterized by the solution size $k$ admits a very simple (and well-known) kernelization algorithm due to Buss and Goldsmith [BG93]: Every vertex that is adjacent to at least $k+1$ other vertices must be contained in a vertex cover of size at most $k$; otherwise all its $k+1$ neighbours must be in a solution. Hence, given an instance $(G, k)$ of Vertex Cover parameterized by the solution size $k$, where $G = (V, E)$ is a graph, we can reduce the instance as long as there exists a vertex of degree at least $k+1$ by deleting this vertex (as well as its incident edges) and by decreasing $k$ by one. Since these vertices must be contained in a solution of size at most $k$ this leads to an equivalent instance. If at some point the value $k$ is smaller than zero, we can return a trivial no-instance. Otherwise, we have reduced the instance $(G, k)$ to an instance $(G', k')$ where every vertex has degree at most $k' \leq k$. Thus, every vertex can cover at most $k'$ edges which implies that we can return a trivial no-instance when $G'$ has more than $k'^2$ edges. Finally, we delete all isolated vertices, i.e., vertices that are not incident with any edge. The resulting instance has at most $k'^2$ edges and at most $2k'^2$ vertices because every vertex is incident with at least one edge.

Next, let us consider the (parameterized) problem $k$-Path[6], which is known to be fixed-parameter tractable [FLS14]. We will explain intuitively why it seems to be unlikely that $k$-Path admits a polynomial kernel parameterized by the solution size $k$ (see also [BDFH09, CFK+15]). Assume that $k$-Path admits a polynomial kernelization algorithm that reduces the number of vertices of a given graph $G = (V, E)$ to at most $k^c$ vertices, where $c$ is a fixed constant, i.e., the algorithm returns a reduced instance $G' = (V', E')$ with $|V'| \leq k^c$. Now, an instance $G$ of $k$-Path could be the disjoint union of $k^{2c+1}$ graphs $G_1, G_2, \ldots, G_{k^{2c+1}}$. Thus, $G$ contains a path of length $k$ if and only if at least one of the graphs $G_1, G_2, \ldots, G_{k^{2c+1}}$ contains a path of length $k$. Applying the kernelization algorithm to the graph $G$ we obtain a reduced instance $G' = (V', k')$ that contains at most $k^c$ vertices. Such an instance can be encoded using $k^{2c}$ bits. Since the number of bits is smaller than the number of graphs, the kernelization algorithm has to discard at least one graph, i.e., has do decide for at least one instances that this instance contains or does not contains a path of length $k$. This should be impossible since $k$-Path is an NP-complete problem and therefore, is an evidence that it is unlikely that $k$-Path has a polynomial kernel.

Overall, parameterized complexity, and more precisely, kernelization allows us to measure the efficiency of data reductions. Furthermore, we are also able to tell when it is unlikely that a (parameterized) problem admits a polynomial kernelization. In this thesis, we focus on kernelization of three different graph problems, namely Vertex Cover, Edge Dominating Set, and Subset Feedback Vertex Set. We will show positive as well as negative results regarding the existence of polynomial kernelizations.

## Thesis Overview

This thesis consists of five parts. Part I contains this introduction as well as the necessary notations and definitions used during this thesis. In Part II we consider the Vertex Cover problem with respect to different structural parameters. Following, in Part III we try to obtain similar results for the Edge Dominating Set problem as for the Vertex Cover problem. We discuss graph cut problems and feedback problems, especially the Subset Feedback Vertex Set problem in Part IV. Finally, in Part V we conclude the work done within this thesis.

In Part I Chapter 2 we reintroduced the concept of graphs and linear programming, related concepts, like orientations, matchings and half-integrality, and some well-known theorems. Furthermore, we give an introduction to computational complexity as well as parameterized complexity. Whereby, the main focus of this thesis is on parameterized complexity.

In Part II we consider the Vertex Cover problem which is one of the most studied problems in parameterized complexity. The Vertex Cover problem is well studied under various structural parameters, i.e., parameters that are independent of the solution size. There is already a quite good knowledge about the line between the existence

[6] The $k$-Path problem, asks whether a given graph $G$ contains a path of length $k$.

of polynomial kernels and kernel lower bounds. Therefore, we start in Chapter 3 with an overview of the known results, especially kernelizations, for Vertex Cover parameterized by different (structural) parameters. Our attention is focused on parameterization by the size of a set $X$, called *modulators*, such that $G - X$ belongs to a given graph class $\mathcal{C}$. The goal of this part is to understand where positive and negative results for polynomial kernelizations for Vertex Cover come from when parameterized by the size of a modulator. We start in Chapter 4 by showing that so-called *blocking sets* play an important role concerning the existence of polynomial kernelization algorithms. In Chapter 5 we consider graph classes that are more general than graph classes $\mathcal{C}$ for which we know that Vertex Cover parameterized by the size of a modulator to graph class $\mathcal{C}$ has a polynomial kernel. Furthermore, some of the graph classes that we consider generalize two incomparable graph classes. We will bound the largest minimal blocking set size of these graph classes. Afterwards, in Chapter 6 we use our knowledge of the previous chapters to obtain, for example, a polynomial kernel for Vertex Cover when parameterized by a parameter that generalizes the parameters feedback vertex set and modulator to treedepth $d$. Finally, we conclude this part in Chapter 7. The results of Part II "Vertex Cover" are based on joint work with Stefan Kratsch [HK17] as well as on joint work with Stefan Kratsch and Astrid Pieterse [HKP20].

In Part III we consider a different graph problem, namely Edge Dominating Set. This problem is in a way a generalization of Vertex Cover. Instead of finding a set of vertices whose deleting results in an independent set, we try to find a set of pairs of vertices whose deletion results in an independent set. Here, the pair of vertices correspond to the set of edges. Besides that, for Edge Dominating Set parameterized by the solution size we can obtain similar results for fpt-algorithms and polynomial kernelizations as for Vertex Cover parameterized by the solution size. We would like to know whether some of the positive results for Vertex Cover parameterized by structural parameters carry over to Edge Dominating Set. Unfortunately, we will show that the Edge Dominating Set problem is much more complicated when it comes to structural parameters than Vertex Cover. We start in Chapter 8 with a short introduction to Edge Dominating Set, by summarizing known results. Moreover, we already consider a few structural parameters for Edge Dominating Set where the kernel lower bound follows easily from known results. Afterwards, we consider in Chapter 9 some "classical" structural parameters, like the size of a feedback vertex set. Unfortunately, we show, among other things, that Edge Dominating Set parameterized by the size of a modulator to well-studied graph classes, like (linear) forests, has no polynomial kernel (unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$). Besides, we show that Edge Dominating Set parameterized by the size of a modulator to a graph class, where every vertex has degree at most one, has a polynomial kernel. We will point out that even constant-size components permitted in $G-X$ seem to behave in a nontrivial way regarding kernelization by $|X|$. This leads to Chapter 10 where we give a complete classification for Edge Dominating Set parameterized by the size of a modulator to a class $\mathcal{C}$ that contains only constant size components. Depending on the components that are contained in $\mathcal{C}$ we show that Edge Dominating Set parameterized by the size of a modulator to class

$\mathcal{C}$ has a polynomial kernel or not. We conclude this part in Chapter 11 with a summary of our results and some open problems. The results of Part III "Edge Dominating Set" are based on joint work with Stefan Kratsch [HK19].

In Part IV, we consider graph cut problems and feedback problems, especially, the Subset Feedback Vertex Set problem, short Subset FVS, which generalizes the well-known Feedback Vertex Set problem. Graph cut problems as well as feedback problems are an interesting part of parameterized complexity. Many people tried to show that, for example, Multiway Cut or the directed version of Feedback Vertex Set, called Directed Feedback Vertex Set have or have not polynomial kernels. This is still unknown and one of the most interesting open problems in parameterized complexity. In Chapter 12, we give an introduction in graph cut problems as well as feedback problems. We will show how these problems relate and will summarize known results regarding fpt-algorithms and kernelization algorithms. Furthermore, we will consider two techniques. One technique uses so-called *important separators* which were introduced by Marx [Mar06] and the other is based on matroids which was first used by Kratsch and Wahlström [KW12]. So far, these matroid based tools are the most promising approach to obtain polynomial kernels for graph cut problems and feedback problems. In Chapter 13 we use the matroid based tools to show that Subset FVS parameterized by the solution size $k$ has a randomized polynomial kernel. Since there is a simple polynomial parameter transformation from Deletable Terminal Multiway Cut to Subset FVS, and Deletable Terminal Multiway Cut parameterized by the solution size $k$ admits a polynomial kernel, it makes sense to study kernelization algorithms for the Subset FVS problem next. Finally, we conclude in Chapter 14. The results of Part IV "Subset FVS" are based on joint work with Stefan Kratsch [HK18].

# CHAPTER 2 PRELIMINARIES

In this chapter we introduce the basic notation and some basic definitions that we need through this thesis. The section about graph theory as well as the section about Linear Programming follows Diestel [Die12] and Korte and Vygen [KV12]. The section about computational complexity follows Arora and Barak [AB09], Garey and Johnson [GJ79], and Korte and Vygen [KV12]. The finial section about parameterized complexity follows Cygan et al. [CFK$^+$15] and Flum and Grohe [FG06].

## 2.1. General Notation

By $\mathbb{N}$ we denote the set of positive integers, and by $\mathbb{N}_0$ we denote the set of non-negative integers, i.e., $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For a positive integer $n \in \mathbb{N}$ we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. An *alphabet* $\Sigma$ is a non-empty finite set of symbols, e.g., numbers and/or letters. A *string* over $\Sigma$ is a finite sequence of elements from $\Sigma$. The length of a string $s$, denoted by $|s|$, is the number of symbols in $s$. By $\Sigma^*$ we denote the set of all strings over $\Sigma$ of any length. A *language* over $\Sigma$ is a subset of $\Sigma^*$. By $\bar{L}$ we denote the *complement* of language $L$ which is defined as $\bar{L} = \Sigma^* \setminus L$.

Let $A$ and $B$ be two sets. The *cartesian product* of the two sets $A$ and $B$, denoted $A \times B$, is the set of all ordered pairs $(a, b)$ with $a \in A$ and $b \in B$. For a positive integer $n \in \mathbb{N}$ the $n$-ary cartesian power of a set $A$ is the set $\{(a_1, a_2, \ldots, a_n) \mid \forall i \in [n] \colon a_i \in A\}$ and we denote this set by $A^n$. The *power set* of $A$, denoted by $2^A$, is the set of all possible subsets of $A$, e.g., the power set of $A = \{1, 2\}$ is $\{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$. Let $n \in \mathbb{N}_0$ be a non-negative integer. By $\binom{A}{n}$ we denote the set of all subsets of $A$ of size $n$ and by $\binom{A}{\leq n}$ we denote the set of all subsets of $A$ of size at most $n$.

Let $A$ be a non-empty set and let $A_1, A_2, \ldots, A_\ell$ be non-empty subsets of $A$. We say that the sets $A_1, A_2, \ldots, A_\ell$ *cover* $A$, if every element of $A$ is contained in at least one of these subsets, i.e., $A = \bigcup_{i=1}^{\ell} A_i$. Furthermore, we say that the non-empty sets $A_1, A_2, \ldots, A_\ell \subseteq A$ are a *partition* of $A$, if every element of $A$ is contained in exactly one of these subsets, i.e., $A = \dot{\bigcup}_{i=1}^{\ell} A_i$. The partition $A_1 = A$ of $A$ is called *trivial partition* of $A$. We call a partition *non-trivial* if it is not trivial.

A set $A$ with some property is said to be a minimal (or maximal) set with this property, if no proper subset (or superset) of $A$ has this property. A set $A$ with some property is said to be a minimum (or maximum) set with this property, if no set with this property has strictly smaller (or strictly larger) cardinality than $A$.

Let $f(n)$ and $g(n)$ be two functions. We write that $f = \mathcal{O}(g)$ if there are two constants $c, n_0 > 0$ such that for all $n > n_0$ it holds that $f(n) \leq c \cdot g(n)$. Furthermore, we write $f = \mathcal{O}^*(g)$ if there exists a constant $c > 0$ such that $f(n) = \mathcal{O}(g(n) \cdot n^c)$.

## 2.2. Graph Theory

We use standard graph notation, mostly following Diestel [Die12] and Korte and Vygen [KV12]. An *undirected graph* is a triple $(V, E, \psi)$, where $V$ and $E$ are finite sets, and $\psi\colon E \to \binom{V}{2} \cup V$. A *directed graph* or *digraph* is a triple $(V, E, \psi)$, where $V$ and $E$ are finite sets and $\psi\colon E \to V \times V$. The *order* of a directed or undirected graph $(V, E, \psi)$ is the cardinality of the set $V$. The set $V$ is called *vertex set*, the elements of $V$ are called *vertices*, the set $E$ is called *edge set* and the elements of $E$ are called *edges*. We call two edges $e, e' \in E$ with $\psi(e) = \psi(e')$ *parallel* edges and we call an edge $e \in E$ a *loop* if $\psi(e) = v$ (undirected graphs) or $\psi(e) = (v, v)$ (directed graphs). In the directed case, we say that $e = (v, w) \in E$ is a *directed* edge with *tail* $v$ and *head* $w$. A graph (directed or undirected) is called *simple* if it has neither parallel edges nor loops.

For simple graphs (directed or undirected) we can identify each edge $e \in E$ with $\psi(e)$. To simplify notation, we denote a simple graph by $G = (V, E)$ where $E \subseteq \binom{V}{2}$ or $E \subseteq V \times V$ instead. Thus, for an edge $e \in E$ with $\psi(e) = \{u, v\}$ or $\psi(e) = (u, v)$, we write $e = \{u, v\}$ or $e = (u, v)$, respectively. The vertices $u$ and $v$ are called *endpoints* of edge $e$. However, we also use this simple notation even when the directed or undirected graph $G$ is not simple. In this case the "set" $E$ may contain several identical elements. We say that an edge $e \in E$ has *multiplicity* $\ell$ when $E$ contains exactly $\ell$ copies of $e$.

We denote the vertex set of a graph $G$ by $V(G)$ and the edge set by $E(G)$. This notation is independent of the name, i.e., for a graph $H = (U, F)$ we denote the vertex set by $V(H)$ and the edge set by $E(H)$. Let $G = (V, E)$ be a directed or undirected graph. A vertex $v \in V$ is *incident* with an edge $e \in E$ if $v$ is an endpoint of $e$. An $e \in E$ is *incident* with an edge $e' \in E$ if the edges $e$ and $e'$ share an endpoint. Two vertices $u, v \in V$ are called *adjacent* if there exists an edge in $E$ that has $u$ and $v$ as its endpoints, i.e., if $\{u, v\} \in E$, $(u, v) \in E$ or $(v, u) \in E$.

Let $G$ and $H$ be two graphs that are either both directed or undirected. We say that $G$ and $H$ are *isomorphic*, denoted by $G \cong H$, if there exists a bijection $\phi\colon V(G) \to V(H)$ such that for all vertices $x, y \in V(G)$ it holds that $\{x, y\} \in E$ or $(x, y) \in E$ if and only if $\{\phi(x), \phi(y)\} \in E$ or $(\phi(x), \phi(y)) \in E$, depending whether $G$ is undirected or directed.

**Edges.** Let $G = (V, E)$ be a directed or undirected graph and let $X, Y \subseteq V$ be two vertex sets. We define $E_G(X, Y) = \{\{x, y\} \in E \mid x \in X, y \in Y\}$ to be the set of edges in $G$ that have one endpoint in $X$ and the other in $Y$, when $G$ is undirected, and $E_G^+(X, Y) = \{(x, y) \in E \mid x \in X, y \in Y\}$ to be the set of edges in $G$ whose tail is in $X$ and whose head is in $Y$, when $G$ is directed. Instead of $E_G(\{x\}, Y)$, $E_G(X, \{y\})$, $E_G^+(\{x\}, Y)$ and $E_G^+(X, \{y\})$ we simply write $E_G(x, Y)$, $E_G(X, y)$, $E_G^+(x, Y)$ and $E_G^+(X, y)$, respectively. Additionally, we write $E_G[X]$ instead of $E_G(X, X)$ and $E_G^+[X]$ instead of $E^+(X, X)$.

For an undirected graph $G = (V, E)$ and a vertex set $X \subseteq V$ we denote the set of edges that are adjacent to exactly one vertex in $X$ by $\delta_G(X)$, i.e., $\delta_G(X) = E(X, V \setminus X)$. For a directed graph $G = (V, E)$ and a vertex set $X \subseteq V$ we denote the set of outgoing or ingoing edges of $X$ by $\delta_G^+(X)$ or $\delta_G^-(X)$, respectively, i.e., $\delta_G^+(X) = E_G^+(X, V \setminus X)$ and $\delta_G^- = E^+(V \setminus X, X)$. For singletons $v \in V$ we write $\delta_G(v)$, $\delta_G^+(v)$, $\delta_G^-(v)$ instead of $\delta_G(\{v\})$, $\delta_G^+(\{v\})$, $\delta_G^-(\{v\})$, respectively. Let $F \subseteq E$ be a set of edges. We call an edge $e \in F$ and $F$-edge. For a set $F \subseteq E$ of edges let $V(F) = \{v \in V \mid \exists e \in F\colon v \in e\}$ be the set of vertices that are incident with at least one edge in $F$.

**Vertices.** Let $G = (V, E)$ be an undirected graph. For a set $X \subseteq V$, let $N_G(X)$ denote the *(open) neighborhood* of $X$ in $G$, i.e., $N_G(X) = \{v \in V \setminus X \mid \exists u \in X\colon \{u, v\} \in E\}$ and let $N_G[X]$ denote the *closed neighborhood* of $X$ in $G$, i.e., $N_G[X] = N_G(X) \cup X$. Again, if $X = \{x\}$ then we write $N_G(x)$ and $N_G[x]$ instead of $N_G(\{x\})$ and $N_G[\{x\}]$, respectively. We call a vertex $v \in N(x)$ a neighbor of $x$. The *degree* of a vertex $v \in V$ is $|\delta_G(v)|$, the number of edges incident with $v$. Observe, that $|\delta_G(v)| = |N_G(v)|$ when $G$ is a simple graph.

Now, let $G = (V, E)$ be a directed graph. The *(open) out-* and *in-neighborhood* of a set $X \subseteq V$ is defined as $N_G^+(X) = \{v \in V \setminus X \mid \exists u \in X\colon (u, v) \in E\}$ and $N_G^-(X) = \{v \in V \setminus X \mid \exists u \in X\colon (v, u) \in E\}$, respectively. As in the undirected case, we define the *closed out-* and *in-neighborhood* of $X \subseteq V$ and we simplify the notation when $X$ contains only one vertex. The *degree* of a vertex $v \in V$ is $|\delta_G^+(v)| + |\delta_G^-(v)|$, the sum of the outgoing and ingoing edges incident with $v$, and the *out-degree* (respectively *in-degree*) of a vertex $v$ is $|\delta_G^+(v)|$ (respectively $|\delta_G^-(v)|$).

To simplify notation, we omit the subscript $G$ whenever the underlying graph is clear from the context.

**Subgraphs.** Let $G = (V, E)$ be a graph, let $X \subseteq V$ and let $F \subseteq E[X]$. The graph $H = (X, F)$ is called *subgraph* of $G$. We call $H$ an *induced* subgraph of $G$ if $E(H) = E_G^{(+)}[X]$ and we denote the induced subgraph of $G$ on vertex set $X$ by $G[X]$. Furthermore, we shorthand $G - X$ for the induced subgraph $G[V \setminus X]$ of $G$, i.e., $G - X$ is the graph obtained from $G$ by deleting $X$ and all edges that are incident with a vertex in $X$, and $G - F$ for the subgraph $(V(G), E(G) \setminus F)$ of $G$. If $X = \{x\}$ then we may also write $G - x$ instead of $G - \{x\}$. Note that the graph $(G - X) - F$ is the same graph as the graph $(G - F) - X$ and we will drop the parentheses.

**Paths and Cycles.** Let $G = (V, E)$ be an undirected or directed graph. A sequence $W = v_1 e_1 v_2 \ldots v_k e_k v_{k+1}$ with $k \geq 0$ such that $e_i = \{v_i, v_{i+1}\} \in E$ or $e_i = (v_i, v_{i+1}) \in E$ for all $i \in [k]$ is called a *walk*. We call the vertices $v_1$ and $v_{k+1}$ *endpoints* of walk $W$ and say that $W$ is a walk *from* $v_1$ *to* $v_{k+1}$. Note that for directed graphs a walk $W$ from $v_1$ to $v_{k+1}$ does not imply a walk from $v_{k+1}$ to $v_1$, whereas for undirected graphs a walk from $v_1$ to $v_{k+1}$ implies that there is also a walk from $v_{k+1}$ to $v_1$. The *length* of walk $W$, denoted by $|W|$, is the number of edges $k$. We say that $W$ has *odd* respectively *even* length if $k$ is odd respectively even. A walk $W$ is *closed* if its endpoints are equal, i.e., $v_1 = v_{k+1}$. We call a walk $W$ a *trail* if in addition all edges are different, i.e., if $e_i \neq e_j$ for all $1 \leq i < j \leq k$.

A *path* is a walk $P = v_1 e_1 v_2 \ldots v_k e_k v_{k+1}$ where additionally all vertices are different, i.e., $v_i \neq v_j$ for all $1 \leq i < j \leq k+1$. Path $P$ is also called a path *from* $v_1$ *to* $v_{k+1}$ or a $v_1, v_{k+1}$*-path*. Again, a $v_1, v_{k+1}$-path in a directed graph does not imply the existence of a $v_{k+1}, v_1$-path. As for walks, the vertices $v_1$ and $v_{k+1}$ are called endpoints of $P$. Furthermore, we call the vertices $v_2, v_3, \ldots, v_k$ *internal* vertices of $P$. For two sets $A, B \subseteq V$ a path with endpoints in $A$ and internal vertices not in $A$ is called an $A$*-path* and a path with one endpoint in $A$, one endpoint in $B$ and internal vertices neither in $A$ nor $B$ is called an $A, B$*-path*. Two path $P_1$ and $P_2$ are *edge-disjoint* or *vertex-disjoint* if they do not share any edge or vertex, respectively. Furthermore, we call two path $P_1$ and $P_2$ *internally vertex-disjoint* if none of them contains an internal vertex of the other. We can extend these definitions to a family $P_1, P_2, \ldots, P_\ell$ of paths. We say that the paths $P_1, P_2, \ldots, P_\ell$ are *pairwise* edge-disjoint, vertex-disjoint or internally vertex-disjoint if all $P_i$, $P_j$ with $1 \leq i < j \leq \ell$ are edge-disjoint, vertex-disjoint or internally vertex-disjoint, respectively.

A *cycle* is a closed walk $C = v_1 e_1 v_2 \ldots v_k e_k v_1$ such that $v_i \neq v_j$ for all $1 \leq i < j \leq k$ and with $k \geq 1$. The length of a path or a cycle is equal to the number of its edges. We call a cycle of odd (even) length also an *odd* (*even*) cycle. Observe that simple graphs cannot have a cycle of length one and only directed simple graphs can have a cycle of length two.

We call a path $P = v_1 e_1 v_2 \ldots v_k e_k v_{k+1}$ or cycle $C = v_1 e_1 v_2 \ldots v_k e_k v_1$ *induced* if for all indices $i, j \in [k+1]$ or $i, j \in [k]$, respectively, with $|i - j| > 1$ it holds that $\{v_i, v_j\} \notin E$, when $G$ is undirected, and $(v_i, v_j) \notin E$, when $G$ is directed. Observe, if $G$ is a simple graph, then for each walk $W = v_1 e_1 v_2 \ldots v_k e_k v_{k+1}$ (and therefore also for each trail, path and cycle) the edge $e_i$ is unique. This implies that $W$ is uniquely described by the sequence $v_1 v_2 \ldots v_k v_{k+1}$. Thus, to simplify notation, we write $W = v_1 v_2 \ldots v_{k+1}$ for a walk, tail, path and cycle instead of $W = v_1 e_1 v_2 \ldots v_k e_k v_{k+1}$. We also use this simple notation for paths, cycles and walks in graph that are not simple when we do not care which edge between two vertices is used by the path, cycle or walk.

**Connectivity and Cuts.** An undirected graph $G = (V, E)$ is *connected* if for each pair $u, v \in V$ of vertices there exists a $u, v$-path in $G$. We call a graph that is not connected *disconnected*. Let $G = (V, E)$ be a directed graph. The *undirected underlying* *graph* of $G$ is obtained from $G$ by replacing all directed edges in $E$ with undirected edges. A directed graph is *connected* if the undirected underlying graph of $G$ is connected. Furthermore, a directed graph $G = (V, E)$ is *strongly connected* if for each pair $u, v \in V$ of vertices there exists a $u, v$-path and a $v, u$-path in $G$. A *connected component* of a directed or undirected graph is a maximal connected subgraph of $G$. We say that a vertex set $X$ is *connected* if $G[X]$ is connected where $G$ is a directed or undirected graph. We call a connected component of $G$ that consists of only one vertex $v$ a *singleton component* and we say that $v$ is an *isolated vertex*.

Let $u, v$ be two different vertices of a directed or undirected graph $G$. We say that $u$ is *reachable* from $v$ or that $v$ *reaches* $u$ if there exists a $v, u$-path in $G$. We say that an edge set $F$ or a vertex set $X$ separates $u$ from $v$ if $u$ is not reachable from $v$ in $G - F$ or $G - X$, respectively. If $G$ is an undirected graph then we only say that the edge set $F$ or the vertex set $X$ separates $u$ and $v$ because the order is not important.

Let $G = (V, E)$ be an undirected graph. A *cut* in $G$ is an edge set $\delta(X)$ for some non-empty set $X \subsetneq V$. We call an edge set $F$ an $(S, T)$*-cut* in $G$ if there exists no $S, T$-path in $G - F$. When $S = \{s\}$ or $T = \{t\}$ we write $(s, T)$-cut, $(S, t)$-cut or $(s, t)$-cut instead of $(\{s\}, T)$-cut, $(S, \{t\})$-cut or $(\{s\}, \{t\})$-cut, respectively. An edge $e \in E$ is called a *bridge* if $G - e$ has more connected components than $G$.

We call a vertex set $X$ a *vertex-cut* in $G$ if there are two vertices in $V \setminus X$ that are separated in $G - X$. A vertex set $X$ is called an $(S, T)$*-vertex cut* in $G$ if there exists no $S \setminus X, T \setminus X$-path in $G - X$. This implies that $X$ must contain the intersection of $S$ and $T$. As before, we simplify the notation by writing $(s, T)$-vertex cut, $(S, t)$-vertex cut or $(s, t)$-vertex cut instead of $(\{s\}, T)$-vertex cut, $(S, \{t\})$-vertex cut or $(\{s\}, \{t\})$-vertex cut, respectively.

Let $G = (V, E)$ be a directed graph. The edge set $\delta^+(X)$ is called a *directed cut* in $G$ if $\emptyset \neq X \subsetneq V$ and $\delta^-(X) = \emptyset$, i.e., no vertex enters the set $X$. A vertex set $X$ is an *vertex-cut* in $G$ if $X$ separates a vertex $v$ from a vertex $u$ that is reachable from $v$ in $G$. An $(S, T)$-cut or an $(S, T)$-vertex cut in $G$ is an edge set $F$ or a vertex set $X$, respectively, such that there is no $S, T$-path in $G - F$ or $G - X$, respectively. Observe, in contrast to the undirected case, there may be $T, S$-paths in $G - F$ or $G - X$.

**Operations on graphs.** Let $G = (V, E)$ be an undirected graph and let $e = \{x, y\}$ be an edge of $G$. We can modify graph $G$ by *contracting* the edge $e$. The resulting graph is obtained from $G$ by deleting the vertices $x$ and $y$, by adding a new vertex $v_e$, and by replacing each edge $e = \{u, v\} \in E$ with $u \in \{x, y\}$ and $v \in V \setminus \{x, y\}$ by the edge $\{v_e, v\}$. We denote the graph that is obtained by contracting edge $e$ in $G$ by $G/e$. Another graph modification is *subdividing* an edge $e = \{x, y\}$ in $G$. Here, the resulting graph is obtained from $G$ by deleting the edge $e$ and by adding a new vertex $v_e$ to the graph as well as the edges $\{x, v_e\}$ and $\{v_e, y\}$.

Let $G = (V, E)$ be a simple undirected graph and let $X \subseteq V$. The graph $\mathrm{torso}(G, X)$ results from the subgraph $G[X]$ by adding an edge between two vertices $x, y \in X$ if and only if there exists an $x, y$-path in $G$ whose internal vertices are disjoint from $X$.

Marx et al. [MOR13] showed that for two vertices $x, y \in X$ it holds that $C \subseteq X$ is an $(x, y)$-vertex cut in $\text{torso}(G, X)$ if and only if $C$ is an $(x, y)$-vertex cut in $G$. This implies that for each connected component $K$ of $G$ with $K \cap X \neq \emptyset$ the set $\text{torso}(G, K)$ is a connected component of $\text{torso}(G, X)$.

Let $G = (V, E)$ be a simple directed or undirected graph. We define $\bar{G} = (\bar{V}, \bar{E})$ as the *complement* of $G$, that is, $\bar{V} := V$, $\bar{E} := \{\{u, v\} \subseteq \binom{V}{2} \mid \{u, v\} \notin E\}$, when $G$ is undirected, and $\bar{E} := \{(u, v) \in V \times V \mid (u, v) \notin E \text{ and } u \neq v\}$, when $G$ is directed.

**Special graphs and sets.** Let $G = (V, E)$ be an undirected graph. A *vertex cover* of $G$ is a vertex set $X \subseteq V$ such that each edge in $E$ is adjacent to a vertex in $X$. We will use $\text{vc}(G)$ to denote the size of a minimum vertex cover of $G$. We call a vertex set $X \subseteq V$ a *clique* if for every pair of vertices $x, y \in X$ it holds that $\{x, y\} \in E$ and an *independent set* if for every pair of vertices $x, y \in X$ it holds that $\{x, y\} \notin E$. Observe, $X$ is a vertex cover of $G$ if and only if $V \setminus X$ is an independent set of $G$. Furthermore, $X$ is an independent set of $G$ if and only if $X$ is a clique in the complement of $G$.

A directed or undirected graph is called *acyclic* if it does not contain any cycle. An undirected acyclic graph is also called *forest*. The degree one vertices of a forest are called *leaves*. We call a forest *linear* if every vertex has degree at most two, i.e., a linear forest is a disjoint union of paths. A connected forest is called a *tree*.

A tree $T = (V, E)$ is called *rooted* if one vertex $r \in V$ has been designated as the *root*. We denote a rooted tree by $(T, r)$ where $r$ is the root. Let $(T, r)$ be a rooted tree. The *children* of a vertex $v \in V$ are the neighbors of $v$ that are not contained in the unique path from $r$ to $v$. For a vertex $v \in V \setminus \{r\}$ we call the unique neighbor of $v$ that is contained on the $r, v$-path in $T$ the *parent* of $v$. The *height* of a rooted tree $(T, r)$ is the length of the longest path from $r$ to a leaf of $(T, r)$. Now, a forest is *rooted* if each connected component is rooted. Thus, a rooted forest has exactly one root for every connected component. The *height* of a rooted forest is the maximum over all heights of its connected components. The *closure* of a rooted tree $(T, r)$ is the graph $H$ that is obtained from tree $T$ by adding for each vertex $v \in V(T)$ edges between $v$ and all vertices that are contained on the unique $t, v$-path in $T$, i.e., $V(H) = V(T)$ and $E(H) = \bigcup_{v \in V(T)} \{\{v, u\} \mid u \neq v \text{ is contained in the unique } r, v\text{-path in } T\}$. Thus, the closure of a rooted forest is the closure of each connected component.

An undirected graph $G = (V, E)$ is *bipartite* if it does not contain any odd cycle, or equivalent, if there exists a partition $U, W$ of $V$ such that $U$ and $W$ are independent sets in $G$. The vertex sets $U$ and $W$ are called *parts* or *bipartition* of $G$. We call a directed or undirected graph *planar* if we can draw it in the plane without crossing edges. A graph $G$ is a *cluster graph* if every connected component is a clique.

By $P_\ell$ we denote the path on $\ell \geq 1$ vertices. The cycle on $\ell \geq 3$ vertices is denoted by $C_\ell$. Furthermore, we denote by $K_\ell$ the clique of size $\ell \geq 1$. The graph $K_3$ is also called *triangle* and it holds that $K_3 \cong C_3$. A *biclique* or *complete bipartite graph*, denoted by $K_{\ell,k}$, is a bipartite graph with bipartition $X$ and $Y$ of size $\ell \geq 1$ and $k \geq 1$, respectively, such that each vertex in $X$ is adjacent to each vertex in $Y$, i.e.,

$K_{\ell,k} = (X\dot{\cup}Y, E)$ with $|X| = \ell$, $|Y| = k$, and $E = \{\{x,y\} \mid x \in X, y \in Y\}$. The graph $K_{1,\ell}$ is called *star* with $\ell$ leaves.

A vertex set $X \subseteq V$ is called a *feedback vertex set* if $G - X$ is a forest and is called an *odd cycle transversal* if $G - X$ is bipartite. We call an edge set $F \subseteq E$ an *edge dominating set* if each edge in $E \setminus F$ is incident with an edge in $F$ or, equivalent, if $V(F)$ is a vertex cover of $G$. We denote the size of a minimum edge dominating set of a graph $G$ by $\textsc{eds}(G)$. Furthermore, we call an edge set $M \subseteq E$ a matching if no two edges in $M$ share an endpoint. We use $\textsc{mm}(G)$ to denote the size of a maximum matching in $G$.

**Matching.** Let $G = (V, E)$ be an undirected graph and let $M$ be a matching in $G$. We say that a vertex $v \in V$ is *matched* or *saturated* by $M$ if it is incident to an edge in $M$. Otherwise, we say that vertex $v$ is *exposed* by $M$. A matching $M$ is a *perfect matching* if every vertex of $G$ is matched, and a *near-perfect matching* if exactly one vertex in $G$ is not matched. We call a graph $G$ *factor-critical* if $G - v$ has a perfect matching for every vertex $v \in V$.

Let $M$ be a matching of $G$. A path $P = v_1e_1v_2 \dots v_ke_kv_{k+1}$ is called *$M$-alternating* if $\{e_1, e_2, \dots, e_k\}\setminus M$ is a matching, i.e., if the edges belong alternately to the matching and not to the matching. An $M$-alternating path is called *$M$-augmenting* if its endpoints are exposed by $M$, i.e., if $v_1, v_{k+1} \notin V(M)$.

It is well known that the size of a maximum matching is a lower bound for the size of a minimum vertex cover. Graphs where the size of a maximum matching is equal to the size of a minimum vertex cover are called *könig graphs*. König [Kön31] showed that bipartite graphs have this property. There are two classical results about matchings in bipartite graphs, Hall's Theorem and the Hopcroft Karp algorithm.

**Theorem 2.1 (Hall's Theorem [Hal35]).** *Let $G$ be a bipartite graph with bipartition $V_1$ and $V_2$. The graph $G$ has a matching that saturates $V_1$ if and only if for all vertex sets $X \subseteq V_1$ it holds that $|N_G(X)| \geq |X|$.*

The following theorem, due to Hopcroft and Karp [HK73], is in some ways similar to Hall's Theorem. It tells us that we can find a matching that saturates $V_1$ or a subset of $V_1$ that violates the Hall condition. The second part of the theorem is not standard (but well-known).

**Theorem 2.2 ([HK73]).** *Let $G$ be an undirected bipartite graph with bipartition $V_1$ and $V_2$, on $n$ vertices and $m$ edges. Then we can find a maximum matching of $G$ in time $\mathcal{O}(m\sqrt{n})$. Furthermore, in time $\mathcal{O}(m\sqrt{n})$ we can find either a maximum matching that saturates $V_1$ or a set $Z \subseteq V_1$ such that $|N_G(Z)| < |Z|$ and such that there exists a maximum matching $M$ of $G - N_G[Z]$ that saturates $V_1 \setminus Z$.*

The following theorem shows that a graph has a certain structure that describes all maximum matching in a given graph, even if this graph is not bipartite.

**Theorem 2.3 (Gaillai-Edmonds decomposition (cf. [Gal64, LP09])).** *Let $G = (V, E)$ be an undirected graph. Denote by $D$ the set of all vertices that are exposed by at least one maximum matching of $G$, by $A$ the neighborhood of $D$, i.e., $A = N_G(D)$, and by $B$ all remaining vertices, i.e., $B = V \setminus (A \cup D)$. The following holds:*

- *The connected components of $G[D]$ are factor-critical.*
- *The graph $G[B]$ has a perfect matching.*
- *Every maximum matching $M$ of $G$ consists of a perfect matching of $G[B]$ a near perfect matching of the connected components of $G[D]$, and matches all vertices of $A$ to distinct connected components of $G[D]$.*

*We call the triplet $(D, A, B)$ the* Gallai-Edmonds decomposition *of $G$.*

## 2.3. Linear Programming

In this section we mostly follow Korte and Vygen [KV12]. A *linear program* (LP) is defined as follows: Given $n$ variables $x_1, x_2, \ldots, x_n$, $m$ linear inequalities or equalities in these variables, called *constraints*, and a linear *cost function*, the goal is to find values for the variables that *maximize* or *minimize* the value the of cost function, subject to the constraints. We often write a linear program as $\min\{\sum_{i=1}^{n} c_i x_i \mid \forall j \in [n]\colon \sum_{i=1}^{n} a_{i,j} x_i \leq b_j\}$ or $\max\{\sum_{i=1}^{n} c_i x_i \mid \forall j \in [m]\colon \sum_{i=1}^{n} a_{i,j} x_i \leq b_j\}$, where $c_i, b_j, a_{ij}$ with $i \in [n], j \in [m]$ are constants, depending whether we want to minimize or maximize the cost function $\sum_{i=1}^{n} c_i x_i$. A *feasible solution* to one of the above linear programs is an assignment to the variables $x_i$, for all $i \in [n]$, which satisfies the constraints of the linear program. We call a feasible solution that attains the minimum respectively maximum an *optimum solution*. It is well known that one can find an optimum solution of a linear program in polynomial time.

An *integer linear program* is a linear program where we have the extra condition that $x_i \in \mathbb{N}_0$ for all $i \in [n]$. Finding an optimum solution to an integer linear program is NP-hard. A *linear program relaxation* of an integer linear program is the integer linear program without the integrality constraint.

Let us consider the Vertex Cover problem, where given an undirected graph $G = (V, E)$ the task is to find a minimum vertex cover. We can formulate this problem as an integer linear problem as follows:

$$\min\left\{\sum_{v \in V} x_v \mid \forall \{u, v\} \in E\colon x_u + x_v \geq 1 \wedge \forall v \in V\colon x_v \in \{0, 1\}\right\}.$$

The linear program relaxation for the Vertex Cover instance of $G = (V, E)$, denoted by $\mathrm{LP}(G)$, is defined as

$$\min\left\{\sum_{v \in V} x_v \mid \forall \{u, v\} \in E : x_u + x_v \geq 1 \wedge \forall v \in V : 0 \leq x_v \leq 1\right\}.$$

It is well known that for the Vertex Cover problem, there is an optimal feasible solution $x$ to $\mathrm{LP}(G)$ such that $x_v \in \{0, \frac{1}{2}, 1\}$ for all $v \in V(G)$. We call a solution for which this holds a *half-integral* solution. Let $x \in \{0, \frac{1}{2}, 1\}^{|V|}$ be a *half-integral* solution to $\mathrm{LP}(G)$. For all $i \in \{0, \frac{1}{2}, 1\}$ we denote by $V_i^x$ the set of vertices in $V$ with $x_v = i$, i.e., $V_i^x = \{v \in V \mid x_v = i\}$. When $x$ is clear from the context, we omit the superscript $x$. Observe, if $x$ is a feasible half-integral solution to $\mathrm{LP}(G)$, then it holds that $N(V_0^x) = V_1^x$. For a feasible solution $x$ to $\mathrm{LP}(G)$, we use $w(x)$ to denote the value of the objective function, i.e., $w(x) = \sum_{v \in V} x_v$. By $\textsc{lp}(G)$ we denote the value of an optimum solution to $\mathrm{LP}(G)$.

**Theorem 2.4 (Nemhauser-Trotter [NT75]).** *Let $G = (V, E)$ be an undirected graph and let $x$ be an optimum solution to $\mathrm{LP}(G)$, the linear program relaxation for the Vertex Cover instance $G$. Let $x^* \in \{0, \frac{1}{2}, 1\}^{|V|}$ with*

$$x_v^* = \begin{cases} 0 & \text{, if } x_v < \frac{1}{2} \\ \frac{1}{2} & \text{, if } x_v = \frac{1}{2} \\ 1 & \text{, if } x_v > \frac{1}{2} \end{cases}.$$

*It holds that $x^*$ is an optimum solution to $\mathrm{LP}(G)$ and that there exists a minimum vertex cover $S$ of $G$ such that $V_1^{x^*} \subseteq S \subseteq V_{\frac{1}{2}}^{x^*} \cup V_1^{x^*}$.*

## 2.4. Computational Complexity

In this section the basic definitions mostly follow Korte and Vygen [KV12]. The parts about complexity classes and complexity-theoretic assumptions follow Arora and Barak [AB09] as well as Garey and Johnson [GJ79]. A *decision problem* asks if something is true, i.e., the possible answers are "yes" or "no". More formally, given an alphabet $\Sigma$, a *decision problem* is a pair $(I, L)$, where $I$ is a language over $\Sigma$ whose elements are called *instances* and a language $L \subseteq I$, and the task is to decide whether a given instance $x \in I$ is contained in $L$. If $x \in L$, then we say that $x$ is a *yes-instance* of $L$, otherwise, $x$ is a *no-instance* of $L$. Most of the time we choose $I = \Sigma^*$ and say that a *decision problem* is a language $L \subseteq \Sigma^*$ over an alphabet $\Sigma$, and the task is to decide whether a given string $x \in \Sigma^*$ is contained in $L$.

An *optimization problem* asks us to find under all feasible solutions one that minimizes or maximizes some objective function. Formally, an optimization problem $\Pi$ is either a minimization or maximization problem that consists of three parts:

- a set $I \subseteq \Sigma^*$ of instances over an alphabet $\Sigma$
- for each instance $x \in I$ a set $S(x)$ of feasible solutions for $x$
- a function $m$ that assigns for each instance $x \in I$ and each feasible solution $y \in S(x)$ a measurement $m(x, y)$, called *solution value* of $y$

An *optimum solution* of an instance $x \in I$ of the optimization problem $\Pi$ is a feasible solution $y^* \in S(x)$ such that for all $y \in S(x)$ it holds that $m(x, y^*) \leq m(x, y)$ or $m(x, y^*) \geq m(x, y)$, respectively. The task is to find for a given instance $x \in I$ an optimum solution.

Observe, we can easily define a decision version of an optimization problem by adding a bound $m_0$ on the solution value of feasible solutions, i.e., instead of asking for an optimum solution, we asks whether there exists a feasible solution with solution value at most (minimization) or at least (maximization) $m_0$. Obviously, if we can solve an optimization problem in polynomial time, then one can solve the associated decision problem with bound $m_0$ in polynomial time.

**Definition 2.5.** Let $L \subseteq \Sigma^*$ be a decision problem over alphabet $\Sigma$ and let $K \subseteq \Gamma^*$ be a decision problem over alphabet $\Gamma$. A *polynomial-time many-one reduction* from language $L$ to language $K$, denoted by $L \leq_m^p K$, is a polynomial-time computable function $f\colon \Sigma^* \to \Gamma^*$ such that $x \in L$ if and only if $f(x) \in K$.

**Complexity classes.** We denote the class of all decision problems $L$ over a given alphabet $\Sigma$ for which there exists a polynomial-time algorithm by $\mathsf{P}$, i.e., there exists an algorithm $A$ such that for each input $x \in \Sigma^*$ the algorithm $A$ accepts $x$ if and only if $x \in L$. The class of all decision problems where we can verify a given solution in polynomial time is called $\mathsf{NP}$. Hence, a decision problem $L$ over alphabet $\Sigma$ belongs to class $\mathsf{NP}$ when there exists a polynomial-time algorithm $A$ such that for each input $x \in \Sigma^*$ it holds that $x \in L$ if and only if there exists a *solution* or *certificate* $y \in \Sigma^*$, whose length is polynomial bounded in $|x|$, such that algorithm $A$ accepts the input $(x, y)$, and for all $x \in \Sigma^* \setminus L$ and all $y \in \Sigma^*$ of polynomial size the algorithm $A$ rejects $(x, y)$. Obviously, it holds that $\mathsf{P} \subseteq \mathsf{NP}$.

A decision problem $L \subseteq \Sigma^*$ is $\mathsf{NP}$-hard when for each decision problem $K \in \mathsf{NP}$ it holds that $K \leq_m^p L$. If additionally it holds that $L \in \mathsf{NP}$ then we say that $L$ is $\mathsf{NP}$-complete, i.e., $L \in \mathsf{NP}$ and $L$ is $\mathsf{NP}$-hard. The class $\mathsf{coNP}$ is the class of all decision problems $L$ over alphabet $\Sigma$ whose complement $\bar{L}$ is contained in $\mathsf{NP}$, i.e., $\mathsf{coNP} = \{L \subseteq \Sigma^* \mid \bar{L} \in \mathsf{NP}\}$. It holds that $P \subseteq \mathsf{NP} \cap \mathsf{coNP}$.

Let $\mathcal{C}$ be a class of decision problems. The class $\mathcal{C}/\mathsf{poly}$ contains all decision problems $L \subseteq \Sigma^*$ for which there exists a decision problem $K \in \mathcal{C}$ and a function $f\colon \mathbb{N} \to \Sigma^*$, called the *advice*, such that $|f(n)| \leq n^c$ for some fixed $c \in \mathbb{N}$ and for all $x \in \Sigma^*$ it holds that $x \in L$ if and only if $(x, f(|x|)) \in K$.

A decision problem $L \subseteq \Sigma^*$ belongs to the class $\mathsf{BPP}$ if and only if there exists a probabilistic polynomial-time Turing machine $T$ and a constant $\frac{1}{2} < p \leq 1$, such that (i) for all inputs $x \in L$, $T$ accepts $x$ with probability at least $p$ and (ii) for all inputs $x \notin L$, $T$ rejects $x$ with probability at least $p$.

We say that a decision problem $L \subseteq \Sigma^*$ is in $\mathsf{RP}$ if and only if there is a probabilistic polynomial-time Turing machine $T$ such that (i) for all $x \notin L$, $T$ rejects $x$, and (ii) for all $x \in L$, $T$ accepts $x$ with probability at least $\frac{1}{2}$. By this definition, it follows straightforwardly that $\mathsf{P} \subseteq \mathsf{RP} \subseteq \mathsf{NP}$ and $\mathsf{RP} \subseteq \mathsf{BPP}$ [Ko82].

**Complexity-theoretic assumptions.** The question whether $\mathsf{P} = \mathsf{NP}$ is one of the major open questions in computer science. It is widely believed that $\mathsf{P} \neq \mathsf{NP}$, i.e., $\mathsf{P} \subsetneq \mathsf{NP}$. Another complexity assumption is that $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$. This assumption is not as strong as $\mathsf{P} \neq \mathsf{NP}$. However, $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$ implies that $\mathsf{P} \neq \mathsf{NP}$. Furthermore, if $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ then the polynomial hierarchy collapses to the third level [Yap83].

In this thesis we will also obtain results under the assumption that $\mathsf{NP} \neq \mathsf{RP}$. Clearly, if the two often believed conjectures $\mathsf{P} = \mathsf{BPP}$ and $\mathsf{P} \neq \mathsf{NP}$ hold, we must have that $\mathsf{P} = \mathsf{RP}$ and $\mathsf{NP} \neq \mathsf{RP}$. Furthermore, this assumption is weaker than the standard assumption $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$ under which kernelization lower bounds are obtained. Since it is known that $\mathsf{RP} \subseteq \mathsf{P/poly}$ [Adl78, Theorem I], it follows that $\mathsf{RP} = \mathsf{NP}$ would imply $\mathsf{NP} \subseteq \mathsf{P/poly}$, implying $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

## 2.5. Parameterized Complexity

In classical complexity theory we are given a decision problem $\mathcal{Q} \subseteq \Sigma^*$, where $\Sigma$ is an alphabet, and we want to either find an algorithm that solves problem $\mathcal{Q}$ in polynomial time in the input size or we want to show that the problem is $\mathsf{NP}$-hard. Since it is widely believed that $\mathsf{P} \neq \mathsf{NP}$ we can assume that we cannot solve $\mathsf{NP}$-hard problems in polynomial time in the input size. However, to classify a problem as $\mathsf{NP}$-hard is useless in practice because this classification does not tell us anything about the structure of the problem. Furthermore, the hardness of a problem often depends on a specific property of the input instance. A multidimensional analysis can help to get a better understanding of the problem and to find more efficient algorithms. This means that we assign an integer value $\ell$, called *parameter*, to a decision problem $Q$ and that we not only measure time with respect to the input size $|x|$, for $x \in \Sigma^*$, but also with the parameter value $\ell$.

Overall, the basic idea of *parameterized complexity theory*, which was pioneered by Downey and Fellows [DF92b], is to refine the analysis of $\mathsf{NP}$-hard problems. In a sense, it is a variant of exact exponential time algorithms for $\mathsf{NP}$-hard problems. We try to make restrictions on the property that causes hardness in order to subsequently develop algorithms that are fast (polynomial time) when the parameter is small (constant). More precisely, it is of interest to ask whether an $\mathsf{NP}$-hard problem has an algorithm that depends exponential only on the parameter and polynomial on the input size.

In this section we give a basic introduction to parameterized complexity, mostly following Cygan et al. [CFK+15] and Flum and Grohe [FG06].[1]

**Parameterized problem.** There are many possibilities to choose the parameter. It can be any part or property of the input instance of the given problem. However, the parameter should usually have some influence on the computational complexity of the

[1] For a more detailed introduction to the field of parameterized complexity, we refer to the book of Cygan et al. [CFK+15].

problem. For a decision version of an optimization problem the first choice for the parameter is often the bound on the solution value. Another way to choose the parameter is to capture some properties. For graph problems we can choose, for example, the maximum degree, the treewidth, or its distance to a forest as the parameter.

**Definition 2.6.** Let $\Sigma$ be a fixed, finite alphabet. A *parameterized problem* is a language $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$. The second component of an instance $(x, \ell) \in \Sigma^* \times \mathbb{N}$ is called the *parameter*.

While it is customary to denote the parameter value by $k$, in this thesis we will generally use $\ell$ for the parameter value and $k$ for the (desired) solution size.

Sometime, a parameterized problem is defined as a pair $(Q, \kappa)$, where $Q$ is a language over a finite alphabet $\Sigma$ and where $\kappa\colon \Sigma^* \to \mathbb{N}_0$ is a mapping that is polynomial time computable, called *parameterization* (cf. [FG06]). Hence, the parameterization assigns a parameter value to each instance of language $Q$.

It is also possible to parameterize by more than one parameter. In this case we can either choose the sum of all parameters or the maximum value over all parameters as our parameter. However, to obtain a better and finer analysis of the running time, when parameterized by the parameters $\ell_1, \ell_2, \ldots, \ell_j$, we will not state the running time in terms of one of the single parameters $\sum_{i=1}^{j} \ell_i$ or $\max\{\ell_i \mid i \in [j]\}$ but in terms of the parameters $\ell_1, \ell_2, \ldots, \ell_j$ independently.

**Fixed-parameter tractable.** In parameterized complexity we are interested in algorithms that solve an instance $(x, k)$ of a parameterized problem in time $f(\ell) \cdot n^c$, where $c$ is a fixed constant and $f\colon \mathbb{N} \to \mathbb{N}$ is a computable function, because these algorithms run in polynomial time when $\ell$ is a constant. Even though algorithms that solve an instance $(x, k)$ of a parameterized problem in time $n^{f(\ell)}$, where $f\colon \mathbb{N} \to \mathbb{N}$ is a computable function, also have a polynomial running time when $\ell$ is a constant, in practice the first algorithm is preferable already for small values of $\ell$.

**Definition 2.7.** Let $\Sigma$ be a fixed, finite alphabet.

- We say that a parameterized problem $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ is *fixed-parameter tractable (FPT)* if there exists a computable function $f\colon \mathbb{N} \to \mathbb{N}$, a constant $c$ and an algorithm $A$, called *fpt-algorithm*, that on input $(x, \ell) \in \Sigma^* \times \mathbb{N}$ takes time at most $f(\ell) \cdot |x|^c$ and correctly decides whether $(x, \ell) \in \mathcal{Q}$. We denote the class of all parameterized problems that are fixed-parameter tractable by *FPT*.

- A parameterized problem $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ is called *slice-wise polynomial (XP)* if there exist two computable functions $f, g\colon \mathbb{N} \to \mathbb{N}$ and an algorithm $A$ that on a given instance $(x, \ell) \in \Sigma^* \times \mathbb{N}$ takes time at most $f(\ell) \cdot |(x, \ell)|^{g(\ell)}$ and correctly decides whether $(x, \ell) \in \mathcal{Q}$. The complexity class containing all slice-wise polynomial problems is called XP.

**Reductions and Parameterized intractability.** In this paragraph we want to introduce the theory of *parameterized intractability.* More precisely, we want to have an evidence that a given parameterized problem is not fixed-parameter tractable. Obviously, parameterized problems that are already $\mathsf{NP}$-hard for constant parameter value cannot be fixed-parameter tractable, unless $\mathsf{P} = \mathsf{NP}$. These problems are also not contained in XP, unless $\mathsf{P} = \mathsf{NP}$. For example, the $q$-Coloring problem[2] parameterized by the integer $q$ is already $\mathsf{NP}$-hard when $q = 3$ [Sto73].

In classical complexity theory we have polynomial-time many-one reductions from a decision problem $\mathcal{Q}$ to a decision problem $\mathcal{K}$ either to know that $\mathcal{Q} \in P$ when $\mathcal{K} \in P$ or to obtain that $\mathcal{K}$ is $\mathsf{NP}$-hard when $\mathcal{Q}$ is $\mathsf{NP}$-hard. We would like to have something similar for parameterized problems, i.e., a reduction from a parameterized problem $\mathcal{Q}$ to a parameterized problem $\mathcal{K}$ such that if $\mathcal{K}$ is fixed-parameter tractable then $\mathcal{Q}$ is fixed-parameter tractable and such that if it is unlikely that $\mathcal{Q}$ is fixed-parameter tractable then it is unlikely that $\mathcal{K}$ is fixed-parameter tractable. We consider the following standard example to see why polynomial-time many-one reductions are not sufficient. Let $G = (V, E)$ be an undirected graph and let $X \subseteq V$. It is well known that a vertex set $X$ is a vertex cover of a graph $G$ if and only if $V \setminus X$ is an independent set of $G$. However, Vertex Cover parameterized by the solution size $k$ has an $\mathcal{O}^*(2^k)$ time fpt-algorithm, but this algorithm does not lead to an $\mathcal{O}^*(f(k))$ time fpt-algorithm for Independent Set parameterized by the solution size, only to an $\mathcal{O}^*(2^{n-k})$ time algorithm.

**Definition 2.8** (Parameterized reduction)**.** Let $\Sigma$ and $\Gamma$ be two fixed, finite alphabets. Let $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ and let $\mathcal{K} \subseteq \Gamma^* \times \mathbb{N}$ be two parameterized problems. A *parameterized reduction or fpt-reduction* from $\mathcal{Q}$ to $\mathcal{K}$ is a mapping $\pi\colon \Sigma^* \times \mathbb{N} \to \Gamma^* \times \mathbb{N}$ such that for all $(x, \ell) \in \Sigma^* \times \mathbb{N}$ it holds that

(i) $\pi((x, \ell)) = (x', \ell')$ can be computed in time $f(k) \cdot |x|^c$, where $f\colon \mathbb{N} \to \mathbb{N}$ is a computable function, and $c$ is a constant,

(ii) $\ell' \leq g(\ell)$, where $g\colon \mathbb{N} \to \mathbb{N}$ is a computable function, and

(iii) $(x, \ell) \in \mathcal{Q}$ if and only if $(x', \ell') \in \mathcal{K}$.

We write $\mathcal{Q} \leq_{fpt} \mathcal{K}$.

Observe that we can concatenate parameterized reductions, i.e., if $\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3$ are parameterized problems with $\mathcal{Q}_1 \leq_{fpt} \mathcal{Q}_2$ and $\mathcal{Q}_2 \leq_{fpt} \mathcal{Q}_3$ then $\mathcal{Q}_1 \leq_{fpt} \mathcal{Q}_3$. Furthermore, the class FPT is closed under parameterized reductions, i.e., if there exists a parameterized reduction from a parameterized problem $\mathcal{Q}$ to a parameterized problem $\mathcal{K}$ that is contained in FPT then it holds that problem $\mathcal{Q}$ is in FPT.

**Definition 2.9.** Let $\Sigma$ be a fixed, finite alphabet. A parameterized problem $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ is *para*-$\mathsf{NP}$ if there exist an alphabet $\Gamma$, a computable function $f\colon \mathbb{N} \to \Gamma^*$, and a decision

[2] In the $q$-Coloring problem we are given a graph $G$, an integer $q$, and the task is to decide whether there exists a coloring $c\colon V(G) \to [q]$ such that for all $\{u, v\} \in E(G)$ it holds that $c(u) \neq c(v)$.

problem $L \subseteq \Sigma^* \times \Gamma^*$ such that $L \in \mathsf{NP}$ and for all instances $(x, \ell) \in \Sigma^* \times \mathbb{N}$ it holds that $x \in \mathcal{Q}$ if and only if $(x, f(\ell)) \in L$. A parameterized problem $Q$ is *para-*$\mathsf{NP}$*-hard* if for any parameterized problem $\mathcal{K}$ in para-$\mathsf{NP}$ there is parameterized reduction from $\mathcal{K}$ to $\mathcal{Q}$. If the parameterized problem $\mathcal{Q}$ is additionally contained in para-$\mathsf{NP}$ then $\mathcal{Q}$ is *para-*$\mathsf{NP}$*-complete.*

Observe, every parameterized problem that is already $\mathsf{NP}$-hard for a constant parameter value is para-$\mathsf{NP}$-hard, i.e., para-$\mathsf{NP}$-hard problems do not have fpt-algorithms under the assumption that $\mathsf{P} \neq \mathsf{NP}$. As mentioned above, it even holds that para-$\mathsf{NP}$-hard problems are not contained in XP unless $\mathsf{P} = \mathsf{NP}$.

It rises the question whether there are other evidences that a parameterized problem is not fixed-parameter tractable besides being para-$\mathsf{NP}$-hard, i.e., whether there exists a concept that is somehow similar to $\mathsf{NP}$-hardness which we can use to show that it is unlikely for a parameterized problem to be fixed-parameter tractable. Downey and Fellows [DF99] introduced the so-called *W-hierarchy.* They defined the complexity classes $W[t]$, for all $t \in \mathbb{N}$ such that $FPT \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq XP$.[3] For this thesis, the precise definitions of all these classes are not important. For our purposes, it suffices to consider only the class $W[1]$ which we define below. Now, a parameterized problem $\mathcal{Q}$ is called $W[t]$-hard if every problem in $W[t]$ is parameterized reducible to $\mathcal{Q}$, and $W[t]$-complement if it is also contained in $W[t]$. It is widely believed that inclusions in the above hierarchy are proper. Thus, the working hypothesis of parameterized complexity is that no $W[1]$-hard problem admits a fpt-algorithm, i.e., $FPT \neq W[1]$.

MULTICOLORED CLIQUE
**Parameter:** $k$
**Input:** An undirected graph $G = (V, E)$, an integer $k$, and a parition of $V$ into $k$ independent sets $V_1, V_2, \ldots, V_k$ of equal size.
**Question:** Does $G$ have a clique of size $k$?

**Definition 2.10** (cf. [FHRV09])**.** The class $W[1]$ contains all parameterized problems $\mathcal{K}$ which are parameterized reducible to MULTICOLORED CLIQUE parameterized by the solution size $k$.

Observe, by the above definition of $W[1]$ it follows that MULTICOLORED CLIQUE is $W[1]$-complete.

**Kernelization.** Data reduction is often used in practice. The goal is to remove redundant parts from the input, to solve some easy parts of the instance, or to obtain some useful properties. The hope is that one can reduce a given instance to a smaller and easier instance of the same problem in polynomial time. Parameterized complexity introduced the concept of *kernelization* to measure the efficiency of reduction rules.

[3]See [CFK+15, Definition 13.16] for a proper definition of $W[t]$ for $t \in \mathbb{N}$.

**Definition 2.11** (Kernelization)**.** Let $\Sigma$ be a fixed, finite alphabet, let $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem, and let $h\colon \mathbb{N} \to \mathbb{N}$ be a computable function. A *kernelization algorithm* for $\mathcal{Q}$ is an algorithm that on input $(x, \ell) \in \Sigma^* \times \mathbb{N}$, takes time polynomial in $|x| + \ell$ and outputs an instance $(x', \ell') \in \Sigma^* \times \mathbb{N}$ such that:

- $|x'|$ and $\ell'$ are bounded by $h(\ell)$, and
- $(x', \ell') \in \mathcal{Q}$ if and only if $(x, \ell) \in \mathcal{Q}$.

The output instance is called *kernel*, while $f(\ell)$ is called the *size* of the kernel. We say that a parameterized problem admits a *polynomial kernelization* or short *polynomial kernel* if it has a kernelization algorithm where the size of the kernel is a polynomial function of the parameter.

In general, a kernelization algorithm is a series of *reduction rules*. A reduction rule is a polynomial-time algorithm that take as input an instance of a parameterized problem $\mathcal{Q}$ over a fixed, finite alphabet $\Sigma$ and produces an instance of the same parameterized problem, called *reduced instance*. We say that a reduction rule is *safe* if for each input instance $(x, k) \in \Sigma^* \times \mathbb{N}$ the reduced instance $(x', k')$ is contained in $\mathcal{Q}$ if and only if $(x, k)$ is contained in $\mathcal{Q}$. An instance is *reduced* with respect to an reduction rule if this rule does not change the instance any more.

Let $\mathcal{Q}$ be a parameterized problem. It is well known that $\mathcal{Q}$ is fixed-parameter tractable if and only if $\mathcal{Q}$ admits a (not necessarily polynomial) kernel [CFK$^+$15]. Thus, we can also show that a problem is fixed-parameter tractable by giving a kernelization algorithm.

**Lower bounds for kernelization.** To show that a parameterized problem does not have a polynomial kernel, Bodlaender et al. [BDFH09] introduced a composition technique which uses a result due to Fortnow and Santhanam [FS11]. This was later refined by Bodlaender et al. [BJK14] to so-called *cross-compositions* that are a convenient front-end for the seminal kernel lower bound framework. Both are generalizations of the notion of reductions that takes as input multiple instances.

**Definition 2.12** (Polynomial equivalence relation)**.** Let $\Sigma$ be a fixed, finite alphabet. A relation $\mathcal{R} \subseteq \Sigma^* \times \Sigma^*$ is a *polynomial equivalence relation* if equivalence of two strings $x, y \in \Sigma^*$ can be tested in time polynomial in $|x| + |y|$ and if $\mathcal{R}$ partitions any finite set $S \subseteq \Sigma^*$ into a number of classes that is polynomially bounded in the largest element of $S$.

**Definition 2.13** ((OR-)cross-composition [BJK14])**.** Let $L \subseteq \Sigma^*$ be a language over a fixed, finite alphabet $\Sigma$, let $\mathcal{R} \subseteq \Sigma^* \times \Sigma^*$ be a polynomial equivalence relation, and let $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. An *(OR-)cross-composition of $L$ into $\mathcal{Q}$* (with respect to $\mathcal{R}$) is an algorithm that, given $t$ instances $x_1, \ldots, x_t \in \Sigma^*$ of $L$ belonging to the same equivalence class of $\mathcal{R}$, takes time polynomial in $\sum_{i=1}^{t} |x_i|$ and outputs an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ such that the following hold:

- “PB”: The parameter value $k$ is polynomially bounded in $\max_{i=1}^{t} |x_i| + \log t$.
- “OR”: The instance $(y, k)$ is a yes-instance for $\mathcal{Q}$ if and only if *at least one* instance $x_i$ is a yes-instance for $L$.

An *(OR-)cross-composition of $L$ into $\mathcal{Q}$ of cost $f(t)$* instead satisfies “OR” and “CB”:

- “CB”: The parameter value $k$ is bounded by $\mathcal{O}(f(t) \cdot (\max_{i=1}^{t} |x_i|)^c)$, where $c$ is some constant independent of $t$.

If $L$ is NP-hard then both forms of cross-compositions are known to imply lower bounds for kernelizations for $\mathcal{Q}$. Theorem 2.15 additionally builds on Dell and van Melkebeek [DvM14].

**Theorem 2.14 ([BJK14, Corollary 3.6.]).** *If an* NP*-hard language $L$ has a cross-composition to $\mathcal{Q}$ then $\mathcal{Q}$ admits no polynomial kernelization or polynomial compression unless* NP $\subseteq$ coNP/poly.

**Theorem 2.15 ([BJK14, Theorem 3.8.]).** *Let $d, \varepsilon > 0$. If an* NP*-hard language $L$ has a cross-composition into $\mathcal{Q}$ of cost $f(t) = t^{1/d+o(1)}$, where $t$ is the number of instances, then $\mathcal{Q}$ has no polynomial kernelization or polynomial compression of size $\mathcal{O}(k^{d-\varepsilon})$ unless* NP $\subseteq$ coNP/poly.

Another way to show that a parameterized problem does not admit a polynomial kernel is to define an appropriate reduction: Let $\Sigma, \Gamma$ be two fixed, finite alphabets. A *polynomial parameter transformation (ppt)* [BTY11] from a parameterized problem $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ to a parameterized problem $\mathcal{K} \subseteq \Gamma^* \times \mathbb{N}$ is a mapping $\pi\colon \Sigma^* \times \mathbb{N} \to \Gamma^* \times \mathbb{N}$ such that for all $(x, \ell) \in \Sigma^* \times \mathbb{N}$ it holds that

- $\pi((x, \ell)) = (x', \ell')$ can be computed in polynomial time,
- $\ell' \leq p(\ell)$, for some polynomial $p$, and
- $(x, \ell) \in \mathcal{Q}$ if and only if $(x', \ell') \in \mathcal{K}$.

Now, assume that there exists a polynomial-parameter transformation from a parameterized problem $\mathcal{Q}$ to a parameterized problem $\mathcal{K}$. If $\mathcal{K}$ has a polynomial kernel then $\mathcal{Q}$ has a polynomial kernel and if it is unlikely that $\mathcal{Q}$ has a polynomial kernel than it is unlikely that $\mathcal{K}$ has a polynomial kernel. Furthermore, if there is a polynomial-parameter transformation from a parameterized problem $\mathcal{Q}$ to a parameterized problem $\mathcal{K}$ with $p(\ell) = c \cdot \ell$ for a constant $c$, then kernelization lower bounds established for $\mathcal{Q}$ also hold for $\mathcal{K}$ [BJK14, BTY11]. We call such polynomial-parameter transformation where $p(\ell) = c \cdot \ell$ a *linear-parameter transformation.*

**Treedepth.** The *treedepth* of a graph $G = (V, E)$, denoted by $\mathrm{td}(G)$, is defined as

$$\mathrm{td}(G) := \begin{cases} 0 & \text{, if } V = \emptyset \\ \min\{\mathrm{td}(G - v) \mid v \in V(G)\} + 1 & \text{, if } G \text{ is connected} \\ \max\{\mathrm{td}(H) \mid H \text{ connected component of } G\} & \text{, otherwise.} \end{cases}$$

We call a rooted forest $F$ such that $G$ is a subgraph of the closure of $F$ a *treedepth forest* of $G$. Recall that every connected component of the forest $F$ has exactly one root, and that if $G$ is connected then $F$ is also connected. This leads to the following alternative definition of treedepth: The treedepth of a graph $G$ is the minimum possible height of any treedepth forest of $G$ plus one. We say that a treedepth forest $F$ of $G$ is *optimum* if the height of $F$ plus one is equal to the treedepth of $G$.

# Part II.

# Vertex Cover

**– The Gentle Problem –**

# CHAPTER 3

# STRUCTURAL PARAMETERS FOR VERTEX COVER

## 3.1. Introduction

In the Vertex Cover problem we are given an undirected graph $G = (V, E)$, an integer $k$, and the question is whether there exists a set $S \subseteq V$ of at most $k$ vertices such that each edge of $G$ is incident with a vertex of $S$, or, in other words, such that $G - S$ is an independent set. This problem is one of Karp's 21 NP-complete problems [Kar72]. Apart from being a classical problem in computational complexity, Vertex Cover is one of the most studied problems in parameterized complexity [CKX10, GP16, Kra18, Lam11, MRS$^+$11, RRS11]. Whenever one wants to test new perspectives or research directions, Vertex Cover is the common problem to start with [AFN04, CJ03, CDR$^+$03, NR00]. It helped, for example, to find new techniques for polynomial kernelization algorithms (cf. [ACF$^+$04]).

Starting with Vertex Cover parameterized by the solution size $k$, the first fpt-algorithm from the year 1993, due to Buss and Goldsmith [BG93], which runs in time $\mathcal{O}(kn + 2^k \cdot k^{2k+k})$, has been improved over the years [DF92b, BFR98, NR99, SF99, CLJ00]. Currently, the best known fpt-algorithm, due to Chen et al. [CKX10], which was presented in 2006, runs in $\mathcal{O}(1.2738^k + kn)$ time and polynomial space. This algorithm improved the $\mathcal{O}(kn+1.286^k)$ time algorithm from 1999, which also needs only polynomial space [CKJ01] as well as the $\mathcal{O}(kn + 1.2745^k k^4)$ time algorithm from 2004, which needs exponential space [CG05]. Additionally, it is known that Vertex Cover does not admit an $\mathcal{O}^*(2^{o(k)})$ time algorithm unless the Exponential Time Hypothesis[1] fails [IPZ01].

The first kernelization algorithm for Vertex Cover parameterized by the solution size $k$ was developed in 1993 by Buss and Goldsmith [BG93]. This algorithm reduces a given instance $(G, k)$ to an equivalent instance with $\mathcal{O}(k^2)$ vertices. Later, this was

[1] Let $\delta$ be the infimum of the set of constants $c$ for which there exists an algorithm that solves 3-SAT in time $\mathcal{O}^*(2^{cn})$. The Exponential Time Hypothesis is the conjecture that $\delta > 0$.

improved to a linear kernel with at most $3k$ vertices using *crown decompositions* [Fel03]. Furthermore, there exist two linear kernels with at most $2k$ vertices. One uses a result of Nemhauser and Trotter [NT75] about the existence of a half-integral solution to the vertex cover LP [CKJ01] and the other is based on iterative compression and crown decompositions [DFRS04]. Besides these positive results, it was shown that there exists no kernel that reduces to $\mathcal{O}(n^{2-\varepsilon})$ bits for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [DvM14]. Thus, we believe that the current bound of $\mathcal{O}(k^2)$ edges is tight up to $k^{o(1)}$ factors. Moreover, it is unlikely that there exists a kernel that reduces to at most $(2-\varepsilon)k$ vertices for any $\varepsilon > 0$: So far, all reduction rules are approximation-preserving [Nie04], which implies that a kernel with $(2-\varepsilon)k$ vertices (for any $\varepsilon > 0$) would yield a polynomial-time approximation algorithm for Vertex Cover with ratio $(2-\varepsilon)$. But this contradicts the Unique Games Conjecture [KR08]. Nevertheless, there exists a result due to Soleimanfallah and Yeo [SY11] that reduces a given Vertex Cover instance to an instance with at most $2k-c$ vertices for every constant $c$. This was improved by Lampis [Lam11] who showed that one can reduce to $2k - c\log(k)$ vertices for every constant $c$. The drawback of these kernelization algorithms is that their running time depends exponentially on $c$.

Despite Vertex Cover being $\mathsf{NP}$-complete, it is solvable in polynomial time on many graph classes, such as forests, bipartite graphs, and even König graphs. This points out the drawback of choosing the solution size as the parameter, because all these graphs can have arbitrarily large vertex covers. This resulted in the study of so-called *structural parameters*, i.e., a parameter value that is largely independent of the solution size. More precisely, a structural parameter is a function on the input structure rather than the standard output size. Jansen and Bodlaender [JB13] started this line of research by considering Vertex Cover parameterized by the size of a feedback vertex set. In other words, they have chosen as parameter the size of a vertex set $X$ such that deleting this set from the input graph results in a forest, and they showed that Vertex Cover admits a polynomial kernel with $\mathcal{O}(|X|^3)$ vertices.

Vertex Cover has turned out to be one of the most fruitful research subjects with a variety of upper and (conditional) lower bounds subject to different parameters (see Section 3.2). The studied structural parameters roughly fall into three types: For width-parameters like treewidth, pathwidth, and treedepth it is known that there is no polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ and the polynomial hierarchy collapses [BDFH09]. Next, there is parameterization by above lower bounds, i.e., the parameter is the difference between the solution size and a lower bound on the solution size. A natural lower bound for Vertex Cover is, for example, the size of a maximum matching. Hence, one can choose $\ell = k - \textsc{mm}$ as a parameter for Vertex Cover, where mm denotes the size of a maximum matching.

The last, and most studied parameter type with respect to Vertex Cover, is parameterization by the deletion distance to a graph class $\mathcal{C}$, i.e., by the minimum size of a, so-called, *modulator* $X$ such that $G-X$ belongs to graph class $\mathcal{C}$. For fixed-parameter tractability and kernelization of the arising parameterized problem it is necessary that Vertex Cover is tractable on inputs from $\mathcal{C}$. For hereditary classes $\mathcal{C}$, this condition is also sufficient for fixed-parameter tractability but not necessarily for the existence of a polynomial kernelization. Interesting choices for $\mathcal{C}$ are various well-studied hereditary graph classes, like forests, bipartite graphs, or chordal graphs, and graphs of bounded treewidth, bounded treedepth, or bounded degree. As a consequence, this rules out fpt-algorithms, and therefore also polynomial kernels, when $\mathcal{C}$ is the class of planar graphs or graphs of degree at most 3, since Vertex Cover is still NP-hard on planar graphs of maximum degree 3 [GJ77].

Besides playing a central role in parameterized complexity, Vertex Cover finds applications in computational biochemistry [RK00, Ste99], computational biology and bioinformatics [BW06, KLW96, SBS05], computational chemistry [DW96, LG07], electrical engineering [CS93, HP00] and classification methods [GKK14].

**Structure of this Chapter.** In the next section we summarize known lower as well as upper bounds for kernelizations of Vertex Cover under different structural parameters. Afterwards, we consider some more general structural parameters to receive a better understanding of the border between lower and upper bounds. In Section 3.3 we define the concept of *blocking sets* which plays a crucial role for Vertex Cover kernelization and we show some basic facts about blocking sets. Finally, in Section 3.4, we show how the above lower bound parameters $k - \textsc{mm}$, $k - \textsc{lp}$, and $k - (2\textsc{lp} - \textsc{mm})$ relate to the size of a modulator to the class of graphs where $\textsc{vc} = \textsc{mm}$, $\textsc{vc} = \textsc{lp}$, or $\textsc{vc} = 2\textsc{lp} - \textsc{mm}$, respectively.

## 3.2. Structural Parameters

The Vertex Cover problem was successfully studied under various structural parameters. As mentioned before, Jansen and Bodlaender [JB13] were the first to study kernelization for Vertex Cover under different, smaller parameters. Their main result is a polynomial kernelization to instances with $\mathcal{O}(|X|^3)$ vertices when $X$ is a *modulator* to the class of forests. Clearly, the size of $X$ is a lower bound on the vertex cover size (as any vertex cover is a modulator to an independent set), and the solution size $k$ cannot be bounded in terms of $\ell = |X|$ alone because forests already have arbitrarily large minimum vertex covers. Since then, their result has been generalized and complemented in several ways.

One of these generalizations was obtained by Kratsch and Wahlström [KW12] for parameterization by the size of a modulator to a bipartite graph. In fact, they showed an even stronger result: The fact that deciding whether a graph $G$ has a vertex cover of size at most $k$ is trivial when $k$ is lower than the size $\textsc{mm}(G)$ of a maximum matching in $G$, or the size $\textsc{lp}(G)$ of an optimum fractional solution to the vertex cover LP of $G$, has motivated the study of above lower bound parameters like $\ell = k - \textsc{mm}(G)$ or $\ell = k - \textsc{lp}(G)$ [RRS11, CLP+14] (see related work). Kratsch and Wahlström [KW12] showed that Vertex Cover parameterized by these two parameters, as well as by the size of a modulator to a König graph[2], has a randomized polynomial kernel. It is well

[2]A graph $G$ is called *König graph* if $\textsc{mm}(G) = \textsc{vc}(G)$.

known that every bipartite graph is a König graph. Furthermore, for Vertex Cover it holds that the parameters $k - \text{LP}(G)$, $k - \text{MM}(G)$, and the size of a modulator to a König graph are equivalent up to a constant factor [FJR13, MRS$^+$11].[3] The strongest lower bound employed so far is $2\text{LP}(G) - \text{MM}(G)$, and Garg and Philip [GP16] gave an $\mathcal{O}^*(3^{k-(2\text{LP}(G)-\text{MM}(G))})$ time algorithm for Vertex Cover. Kratsch [Kra18] showed that there also exists a randomized polynomial kernel for this parameter.

Majumdar et al. [MRS18] studied Vertex Cover parameterized by the size of a modulator $X$ to a graph of maximum degree at most $d$. For $d \geq 3$ this problem is NP-hard but for $d = 2$ and $d = 1$ they obtained kernels with $\mathcal{O}(|X|^5)$ and $\mathcal{O}(|X|^2)$ vertices, respectively. They extend their idea for Vertex Cover parameterized by the size of a degree-2-modulator to a kernel with $\mathcal{O}(|X|^9)$ vertices where $X$ is a modulator to a graph where every connected component is a tree or a cycle. Their result motivated Fomin and Strømme [FS16] to investigate a parameter that is smaller than both a modulator to a disjoint union of trees and cycles, and the size of a feedback vertex set. They consider $X$ being a modulator to a *pseudoforest*, i.e., with each connected component of $G - X$ having at most one cycle. For this they obtained a kernelization to $\mathcal{O}(|X|^{12})$ vertices, generalizing (except for the size) the results of Majumdar et al. [MRS18] and Jansen and Bodlaender [JB13]. Furthermore, Majumdar et al. [MRS18] also showed that Vertex Cover parameterized by the size of a modulator $X$ to a cluster graph with cliques of size bounded by $d$, admits a kernel with $\mathcal{O}(|X|^d)$ vertices. Additionally, they ruled out kernels of size $\mathcal{O}(|X|^{d-\epsilon})$ for this parameter. Recent work of Bougeret and Sau [BS19] shows that Vertex Cover admits a kernel of size $\mathcal{O}(|X|^{f(c)})$ when $X$ is a modulator to a graph of treedepth at most $c$.

Regarding lower bounds for kernelization (all assuming $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$), it is well known that there are no polynomial kernels for Vertex Cover when parameterized by width-parameters like treewidth, pathwidth, or treedepth (cf. [BDFH09]). Bodlaender et al. [BJK14] showed that there is no polynomial kernelization in terms of the vertex deletion distance to a single clique, which is stronger than deletion distance to cluster or perfect graphs for example. Other lower bounds were obtained by Cygan et al. [CLP$^+$14] for modulators to treewidth at most two and Jansen [Jan13] for modulators to outerplanar graphs. Fomin and Strømme [FS16] showed a lower bound for modulators to mock-forests[4] using a similar construction as Cygan et al. [CLP$^+$14] and Jansen [Jan13]. Figure 3.1 summarizes the known results for Vertex Cover parameterization and shows the hierarchy of these parameters.

**Related work.** It was first shown by Razgon and O'Sullivan [RO09] that Vertex Cover parameterized by $\ell = k - \text{MM}$, where MM stands for the size of a maximum matching and $k$ for the solution size, is solvable in $\mathcal{O}^*(15^\ell)$ time, and is therefore fixed-parameter tractable. In other words, the parameter value $\ell$ is the difference between

[3] It holds that $k - \text{LP}(G) \leq k - \text{MM}(G) \leq |X| \leq 2(k - \text{MM}(G)) \leq 4(k - \text{LP}(G))$, where $X$ is a modulator to a König graph.

[4] A mock-forest is a graph where no two cycles share a vertex.

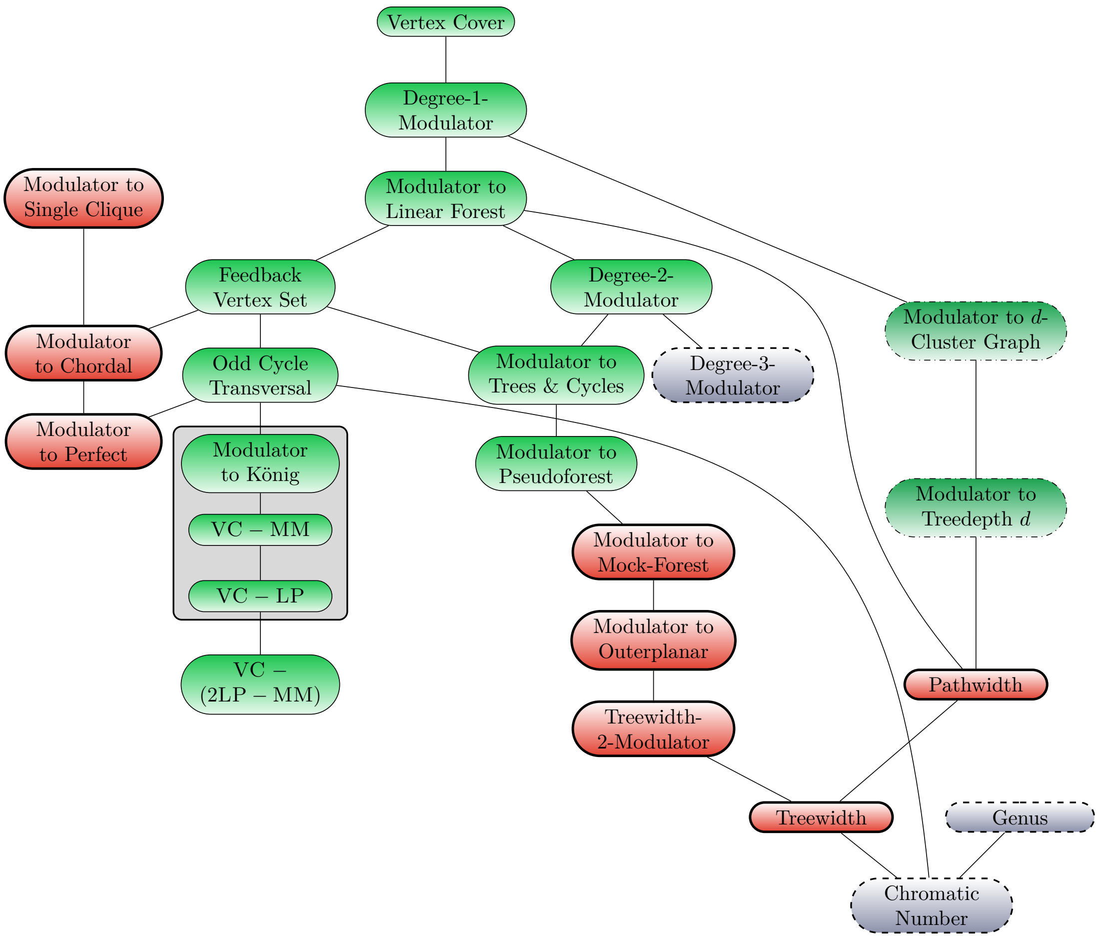



Figure 3.1.: Parameter hierarchy for Vertex Cover, where we assume that the modulator is given in the input. The shadings indicated that the parameterization is either para-NP-complete , FPT but conditionally lacking a polynomial kernel , FPT with a polynomial kernel for constant $d$ , or FPT with polynomial kernel . A line between two parameters indicates that the lower parameter can be bounded in a function of the higher parameter. The three parameters that are grouped are equivalent up to constant factors.

the solution size $k$ and the obvious lower bound MM. This was improved by Raman et al. [RRS11] and later by Cygan et al. [CPPW13a] to an algorithm with running time $\mathcal{O}^*(4^\ell)$. Narayanaswamy et al. [NRRS12] improved this result by showing that Vertex Cover parameterized by $\ell = k - \text{LP}$, where LP stands for the minimum *fractional vertex cover* (as determined by the LP relaxation), can be solved in time $\mathcal{O}^*(2.6181^\ell)$. Currently, the best running time, due to Lokshtanov et al. [LNR+14], is $\mathcal{O}^*(2.3146^\ell)$. As mentioned above, Garg and Philip [GP16] proved that Vertex

COVER is also fixed-parameter tractable even for parameter $\ell = k - (2\textsc{lp} - \textsc{mm})$. Their algorithm runs in time $\mathcal{O}^*(3^\ell)$.

Another lower bound for the size of a minimum vertex cover in a graph $G$ on $n$ vertices with minimum degree 2 and maximum degree $\delta$ is $\frac{2}{2+\delta} \cdot n$ (cf. [Xia10b]). Since it is easy to reduce degree one vertices (in polynomial time), one can assume, without loss of generality, that every input graph has minimum degree 2. Xiao [Xia10b] showed that VERTEX COVER is solvable in time $\mathcal{O}(1.6651^\ell)$ on graphs with maximum degree 3 (and minimum degree 2) where $\ell = k - \frac{2}{2+3} \cdot n$. Recently, Tsur [Tsu18] gave an $\mathcal{O}^*(1.6253^\ell)$ time algorithm for VERTEX COVER on graphs with maximum degree 4 (and minimum degree 2) with $\ell = k - \frac{2}{2+4} \cdot n$.

**Distance to small modulators.** As mentioned above, VERTEX COVER has a polynomial kernel when parameterized by the size of a modulator to a pseudoforest, and no polynomial kernel when parameterized by the size of a modulator to a mock-forest [FS16]. Arguably, there is still quite some room between allowing a single cycle per component and allowing an arbitrary number of cycles as long as they share no vertices. Therefore, we introduce as a structural parameter for VERTEX COVER the size of a modulator to a *d-quasi-forest*. A *d-quasi-forest* is a graph where each connected component has a feedback vertex set of size at most $d$. Similar, one can generalize the parameter size of a modulator to a bipartite graph as well as the parameter $\ell = k - \textsc{lp}$. We say that a graph is *d-quasi-bipartite* if each connected component has an odd cycle transversal of size at most $d$ and we say that a graph $G$ is *d-quasi-integral* if in each connected component the size of a minimum vertex cover exceeds the size of an optimum LP solution by at most $d$, i.e., a graph $G$ is $d$-quasi-integral if for each connected component $H$ of $G$ it holds that $\textsc{vc}(H) \leq \textsc{lp}(H) + d$. Now, the parameter is the size of a modulator to a $d$-quasi-bipartite or $d$-quasi-integral graph.

These new parameters are incomparable to the parameter $k-(2\textsc{lp}-\textsc{mm})$ and the size of a modulator to a graph of treedepth at most $d$: Already the kernelization by the size of a feedback vertex set [JB13], which is already generalized by the size of a modulator to a $d$-quasi-forest, allows arbitrarily long paths in $G - X$. Such paths are forbidden in a graph of bounded treedepth. In addition, a disjoint union of cliques $K_4$ with four vertices is a 2-quasi-forest, where $2\textsc{lp}(K_4) - \textsc{mm}(K_4) = 2$ but the vertex cover size is three per component. Conversely, taking a star with $c$ leaves and appending a 3-cycle at each leaf yields a graph with a feedback vertex set as well as an odd cycle transversal of size equal to $c$, and $k - \textsc{lp} = \frac{c-1}{2}$, but with constant treedepth and $k - (2\textsc{lp} - \textsc{mm}) = 0$; $c$ can be chosen arbitrarily large.

**Elimination distance.** Bulian and Dawar [BD16] introduced the structural parameter *elimination distance to graph class* $\mathcal{C}$ which is a natural generalization of the parameters deletion distance to a class $\mathcal{C}$ and treedepth (see also [Bul17]). The elimination distance to graph class $\mathcal{C}$ is defined in the same way as treedepth except that all graphs from $\mathcal{C}$ get value 0 rather than just the empty graph. Intuitively, elimination distance to $\mathcal{C}$ can be pictured as having a tree-like deletion of vertices (as for treedepth) but being allowed to stop when the remaining connected components belong to $\mathcal{C}$ (rather than continuing to the empty graph).

Now, one can argue whether each connected component of the remaining graph should be contained in $\mathcal{C}$ or the entire graph consisting of the remaining connected components should be contained in $\mathcal{C}$. Bulian and Dawar [BD16, BD17] chose the first version. However, to have a closer relation to the deletion distance to $\mathcal{C}$, the second variant seems to be more natural. Moreover, the second variant is more general: In the first variant, where we want each connected component to belong to $\mathcal{C}$, we should assume that $\mathcal{C}$ is closed under removing connected components. Further, given a graph class $\mathcal{C}$ that is closed under removing connected components, we can define $\mathcal{C}^*$ as the graph class that contains exactly the graphs where each connected component belongs to $\mathcal{C}$. Now, the elimination distance to graph class $\mathcal{C}^*$ using the second variant is equivalent to the elimination distance to graph class $\mathcal{C}$ using the first variant.

Bulian and Dawar [BD16, Bul17] define the elimination distance only for graph classes that are closed under taking subgraphs. However, for their definition it is enough to assume that class $\mathcal{C}$ is closed under removing connected components. For the reasons mentioned above, we prefer the second variant. To this end, we introduce the notion of *robust* graph classes, meaning that the graph class $\mathcal{C}$ is closed under disjoint union and under removing connected components. This allows us to use the same recursive definition of elimination distance as Bulian and Dawar [BD16, BD17] when the graph class is robust (see Definition 3.2).

First of all, we give a definition that extends the parameter elimination distance to arbitrary graph classes $\mathcal{C}$ (that are not necessarily robust):

**Definition 3.1.** Let $\mathcal{C}$ be an arbitrary graph class, and let $G = (V, E)$ be a graph. The *elimination distance of $G$ to graph class $\mathcal{C}$* is the minimum treedepth of $\mathrm{torso}(G, X)$ where $X$ is a modulator to graph class $\mathcal{C}$, i.e.,

$$\mathrm{ed}_{\mathcal{C}}(G) = \min\{\mathrm{td}(\mathrm{torso}(G, X)) \mid X \subseteq V : G - X \in \mathcal{C}\}$$

As mentioned above, under the assumption that graph class $\mathcal{C}$ is robust, we can define the elimination distance of a graph $G$ to graph class $\mathcal{C}$ recursively, similar to the treedepth of a graph (cf. [BD17, Bul17]).

**Definition 3.2.** Let $\mathcal{C}$ be a robust graph class and let $G = (V, E)$ be a graph. We define the *elimination distance to $\mathcal{C}$* as

$$\mathrm{ed}^*_{\mathcal{C}}(G) := \begin{cases} 0 & \text{, if } G \in \mathcal{C} \\ \min\{\mathrm{ed}_{\mathcal{C}}(G - v) \mid v \in V\} + 1 & \text{, if } G \notin \mathcal{C},\ G \text{ connected} \\ \max\{\mathrm{ed}_{\mathcal{C}}(H) \mid H \text{ connected component of } G\} & \text{, otherwise.} \end{cases}$$

**Lemma 3.3.** *Definition 3.1 and Definition 3.2 are equivalent for robust graph classes.*

**Proof.** We prove via induction that $\mathrm{ed}_{\mathcal{C}}(G) \leq \mathrm{ed}^*_{\mathcal{C}}(G)$ and that $\mathrm{ed}^*_{\mathcal{C}}(G) \leq \mathrm{ed}_{\mathcal{C}}(G)$ for every graph $G$ and every robust graph class $\mathcal{C}$. In the first base case we assume that $\mathrm{ed}^*_{\mathcal{C}}(G) = 0$. This implies that $G$ is a graph of graph class $\mathcal{C}$. Thus, for $X = \emptyset$ it holds that $G - X \in \mathcal{C}$ which implies that $\mathrm{ed}_{\mathcal{C}}(G) \leq \mathrm{td}(\mathrm{torso}(G, X)) = 0$ because $\mathrm{torso}(G, X)$ is the empty graph. Now, assume for the second base case that $\mathrm{ed}_{\mathcal{C}}(G) = 0$. Let $X \subseteq V$ such that $G - X \in \mathcal{C}$ and $\mathrm{ed}_{\mathcal{C}}(G) = \mathrm{td}(\mathrm{torso}(G, X))$. Since $\mathrm{td}(H) = 0$ if and only if $V(H) = \emptyset$ it follows that $\mathrm{torso}(G, X)$ is the empty graph. Hence, $X$ is the empty set which implies that $G$ is contained in graph class $\mathcal{C}$. This shows that $\mathrm{ed}^*_{\mathcal{C}}(G) = 0$ and concludes the proof of the base case.

For the induction step, we assume that $\mathrm{ed}_{\mathcal{C}}(G) \leq \mathrm{ed}^*_{\mathcal{C}}(G)$ when $\mathrm{ed}^*_{\mathcal{C}}(G) < i$ and that $\mathrm{ed}^*_{\mathcal{C}}(G) \leq \mathrm{ed}_{\mathcal{C}}(G)$ when $\mathrm{ed}_{\mathcal{C}}(G) < i$, where $\mathcal{C}$ is an arbitrary robust graph class.

Observe, if $\mathcal{C}$ is a robust graph class then it holds that $\mathrm{ed}_{\mathcal{C}}(G) = \max\{\mathrm{ed}_{\mathcal{C}}(H) \mid H \text{ connected component of } G\}$: Since $\mathcal{C}$ is a robust graph class it holds that a modulator to graph class $\mathcal{C}$ in $G$ is the union of modulators to class $\mathcal{C}$ for each connected component of $G$. Together with the fact that $\mathrm{td}(G) = \max\{\mathrm{td}(H) \mid H \text{ connected component of } G\}$ and the fact that there exists a one-to-one correspondence between the connected components of $G$ that contain at least one vertex of $X$ and the connected components of $\mathrm{torso}(G, X)$, it follows that $\mathrm{ed}_{\mathcal{C}}(G) = \max\{\mathrm{ed}_{\mathcal{C}}(H) \mid H \text{ connected component of } G\}$. Thus, it is enough to prove the induction steps for connected graphs.

First, let $\mathcal{C}$ be a robust graph class and let $G$ be any connected graph with $\mathrm{ed}^*_{\mathcal{C}}(G) = i > 0$. We have to show that $\mathrm{ed}_{\mathcal{C}}(G) \leq \mathrm{ed}^*_{\mathcal{C}}(G) = i$. Since $G$ is connected it holds that there exists a vertex $v \in V(G)$ such that $\mathrm{ed}^*_{\mathcal{C}}(G) = \mathrm{ed}^*_{\mathcal{C}}(G - v) + 1$. By the induction hypothesis it holds that

$$\mathrm{ed}^*_{\mathcal{C}}(G - v) \geq \mathrm{ed}_{\mathcal{C}}(G - v) = \min\{\mathrm{td}(\mathrm{torso}(G - v, X)) \mid X \subseteq V(G - v) \colon G - v - X \in \mathcal{C}\}.$$

Let $X$ be a modulator to $\mathcal{C}$ of graph $G - v$ such that $\mathrm{ed}_{\mathcal{C}}(G - v) = \mathrm{td}(\mathrm{torso}(G - v, X))$. It holds that $\mathrm{torso}(G - v, X) = \mathrm{torso}(G, X \cup \{v\}) - v$ by the definition of torso. Overall, by the definition of treedepth, this implies that

$$\begin{aligned}
\mathrm{ed}^*_{\mathcal{C}}(G) &= \mathrm{ed}^*_{\mathcal{C}}(G - v) + 1 \geq \mathrm{ed}_{\mathcal{C}}(G - v) + 1 = \mathrm{td}(\mathrm{torso}(G - v, X)) + 1 \\
&= \mathrm{td}(\mathrm{torso}(G, X \cup \{v\}) - v) + 1 \geq \mathrm{td}(\mathrm{torso}(G, X \cup \{v\})) \\
&\geq \mathrm{ed}_{\mathcal{C}}(G).
\end{aligned}$$

Second, let $\mathcal{C}$ be a robust graph class and let $G$ be any connected graph with $\mathrm{ed}_{\mathcal{C}}(G) = i > 0$. We have to prove that $\mathrm{ed}^*_{\mathcal{C}}(G) \leq \mathrm{ed}_{\mathcal{C}}(G)$. Let $X \subseteq V(G)$ such that $G - X \in \mathcal{C}$ and $\mathrm{ed}_{\mathcal{C}}(G) = \mathrm{td}(\mathrm{torso}(G, X))$. By the definition of treedepth there exists a vertex $x \in X$ such that $\mathrm{td}(\mathrm{torso}(G, X)) = \mathrm{td}(\mathrm{torso}(G, X) - x) + 1$. As before, it holds that $\mathrm{torso}(G, X) - x = \mathrm{torso}(G - x, X \setminus \{x\})$. Since $X \setminus \{x\}$ is a modulator to $\mathcal{C}$ of graph $G - x$ it holds that $\mathrm{ed}_{\mathcal{C}}(G - x) \leq \mathrm{td}(\mathrm{torso}(G - x, X \setminus \{x\})) = \mathrm{ed}_{\mathcal{C}}(G) - 1 = i - 1$. Thus, by the induction hypothesis it holds that $\mathrm{ed}^*_{\mathcal{C}}(G - x) \leq \mathrm{ed}_{\mathcal{C}}(G - x)$. Furthermore, by the definition of $\mathrm{ed}^*_{\mathcal{C}}$ it follows that $\mathrm{ed}^*_{\mathcal{C}}(G) \leq \mathrm{ed}^*_{\mathcal{C}}(G-x)+1$. Overall, we obtain that

$$\mathrm{ed}_{\mathcal{C}}(G) = \mathrm{ed}_{\mathcal{C}}(G-x)+1 \geq \mathrm{ed}^*_{\mathcal{C}}(G-x)+1 \geq \mathrm{ed}^*_{\mathcal{C}}(G).$$

This concludes the proof. ■

By the definition of elimination distance, the treedepth of a graph $G$ corresponds to its elimination distance to the empty graph, and $\mathrm{ed}_{\mathcal{C}_{IS}}(G) = \mathrm{td}(G)+1$ where $\mathcal{C}_{IS}$ is the class of edgeless graphs.

Analogous to the treedepth forest, we can define an *elimination forest*. For a graph $G$, an elimination forest with respect to $\mathcal{C}$ is a treedepth forest of $\mathrm{torso}(G, X)$ where $X \subseteq V(G)$ such that $G - X \in \mathcal{C}$. As before, we say that an elimination forest $F$ is *optimum* if its height plus one is equal to the elimination distance of $G$ to graph class $\mathcal{C}$. By definition, it holds that there always exists an optimum elimination forest. Let $F$ be an elimination forest of $G$ with respect to $\mathcal{C}$, where $\mathcal{C}$ is a robust graph class. We call the connected components of $G - V(F)$ *base components* of $F$. Under the assumption that $\mathcal{C}$ is robust it holds that each base component is contained in $\mathcal{C}$. Furthermore, it holds that every base component of an elimination forest $F$ of $G$ with respect to class $\mathcal{C}$ is only adjacent to vertices of one path from a root to a leaf in $F$. Observe, if $G$ is connected then $F$ is a tree and has exactly one root.

**Notation.** During this part we consider Vertex Cover parameterized by the size of a modulator to a graph class $\mathcal{C}$ and to graphs with bounded elimination distance. Let $\mathcal{C}$ be a graph class, let $G$ be a graph and let $X \subseteq V(G)$. The set $X$ is called a *$\mathcal{C}$-modulator* if $G - X \in \mathcal{C}$, and we call the set $X$ a *$(\mathcal{C}, d)$-modulator* if $\mathrm{ed}_{\mathcal{C}}(G-X) \leq d$. When considering Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator or a $(\mathcal{C}, d)$-modulator, we will assume that this modulator is given on input. As such, inputs to the problem are triplets $(G, k, X)$, where $G$ is a graph, the set $X$ is a modulator in $G$, and the problem is to decide whether $G$ has a vertex cover of size at most $k$.

Let $\mathcal{C}$ be a robust graph class, and let $c$ be a positive integer. By $\mathcal{C} + c$ we denote the graph class that consists of all graphs $G$ where each connected component $H$ of $G$ contains a $\mathcal{C}$-modulator $X_H$ of size at most $c$ such that $H - X_H$ is still connected, i.e.,

$$\mathcal{C} + c := \{G \mid \forall \text{c.c. } H \text{ of } G\, \exists X_H \subseteq V(H), |X_H| \leq c\colon H - X_H \in \mathcal{C}, H - X_H \text{ connected}\}.$$

By definition, the graph class $\mathcal{C}+c$ is still robust. It also holds that $(\mathcal{C}+c)+1 = \mathcal{C}+(c+1)$: Let $G$ be a graph of graph class $(\mathcal{C}+c)+1$, i.e., for each connected component $H$ of $G$ there exists a vertex $v_H$ such that $H - v_H$ is connected and belongs to $\mathcal{C}+c$. Hence, there exists a set $X_H$ of size at most $c$ such that $H - v_H - X_H$ is a connected graph of graph class $\mathcal{C}$. Now, $X_H \cup \{v_H\}$ is a $\mathcal{C}$-modulator of size at most $c+1$ and $H - X_H - v_H$ is connected, thus $G \in \mathcal{C} + (c+1)$.

Conversely, let $G$ be a graph that is contained in graph class $\mathcal{C}+(c+1)$. This implies that for each connected component $H$ of $G$ there exists a $\mathcal{C}$-modulator $X_H$ of size at most $c+1$ such that $H - X_H$ is still connected. Since $H$ is connected there exists a vertex $v_H \in X_H$ such that $H-(X_H \setminus \{v_H\})$ is still connected. The vertex $v_H$ is a $(\mathcal{C}+c)$-modulator of $H$ because $H-(X_H \setminus \{v_H\})$ is connected by the choice of $v_H$, and because $X_H \setminus \{v_H\}$ is a $\mathcal{C}$-modulator of $H - v_H$ of size at most $c$ where $(H-v_H)-(X_H \setminus \{v_H\})$ is still connected.

Observe that we can solve VERTEX COVER in polynomial time on graph class $\mathcal{C}+c$ when VERTEX COVER parameterized by the size of a $\mathcal{C}$-modulator is fixed-parameter tractable. Hence, if VERTEX COVER is polynomial-time solvable on a hereditary graph class $\mathcal{C}$ then VERTEX COVER is polynomial-time solvable on graph class $\mathcal{C}+c$.

Adding the above parameters for VERTEX COVER into Figure 3.1 results in the parameter hierarchy of Figure 3.2. We will show that VERTEX COVER parameterized by these parameters has a polynomial kernel (Chapter 6).

## 3.3. Blocking Sets

In the present work, we seek to unify and generalize existing results about VERTEX COVER kernelization by using so-called *blocking sets*.

**Definition 3.4** (Blocking set)**.** Let $G$ be a graph and let $Y \subseteq V(G)$ be a subset of its vertices. We say that the set $Y$ is a *blocking set* in $G$ if there exists no vertex cover $S$ of $G$ such that $Y \subseteq S$ and $|S| = \text{VC}(G)$. In other words, there is no optimal vertex cover of $G$ that contains $Y$. A blocking set $Y$ is *minimal* if no strict subset of $Y$ is also a blocking set.

It holds that for each graph $G=(V,E)$ the set $V$ itself is a blocking set. But, in general, this set is not a minimal blocking set of $G$. Several graph classes have constant upper bounds on the size of minimal blocking sets, e.g., in any forest (or even in any bipartite graph) every minimal blocking set has size at most two. On the other hand, even restrictive classes like outerplanar graphs have unbounded minimal blocking set size, i.e., for each integer $d$ there is a graph in the class with a minimal blocking set of size greater than $d$. As a final example, cliques are the unique graphs for which $V$ is the only minimal blocking set because all optimal vertex covers have form $V \setminus \{v\}$ for any vertex $v \in V$; in particular, any graph class containing all cliques has unbounded minimal blocking set size.

**Notation.** Let $G$ be a graph, we use $\beta(G)$ to denote the size of the largest minimal blocking set in $G$. For a graph class $\mathcal{C}$, let $\beta_{\mathcal{C}} := \infty$ if the minimal blocking set size of graphs in this graph class can be arbitrarily large and let $\beta_{\mathcal{C}} := \max_{G \in \mathcal{C}} \beta(G)$, otherwise. Define $\beta_{\mathcal{C}}(d) := \max\{\beta(G) \mid \text{ed}_{\mathcal{C}}(G) \le d\}$.

**Basic properties of blocking sets.** First of all, we show some basic facts about minimal blocking sets, as well as some useful connections between minimal blocking sets of a graph $G$ and certain subgraphs of $G$. These properties help us, for example, in Chapter 5 to bound $\beta_{\mathcal{C}}(d)$ for every hereditary graph class $\mathcal{C}$ where $\beta_{\mathcal{C}}$ is bounded.

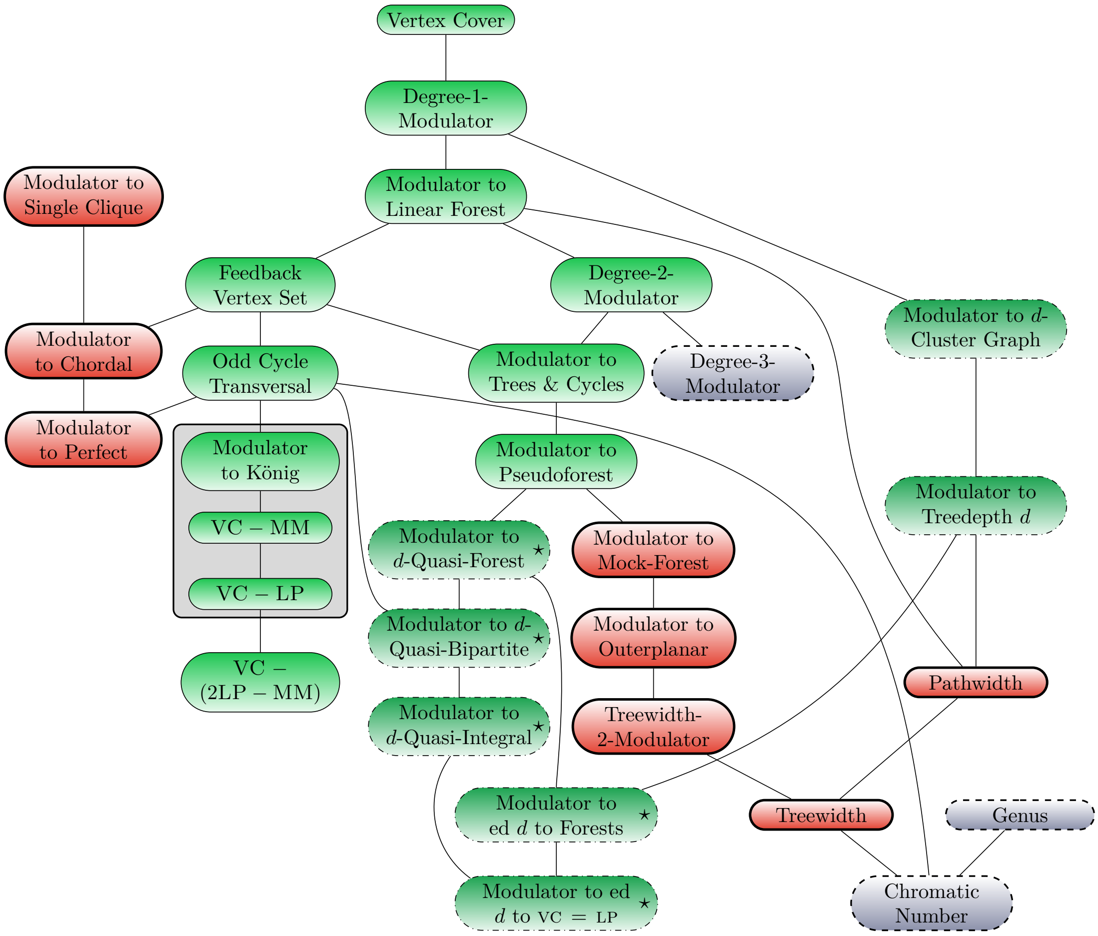



Figure 3.2.: Parameter hierarchy for Vertex Cover, where we assume that the modulator is given in the input. The shadings indicated that the parameterization is either para-NP-complete, FPT but conditionally lacking a polynomial kernel, FPT with a polynomial kernel for constant $d$, or FPT with polynomial kernel. A line between two parameters indicates that the lower parameter can be bounded in a function of the higher parameter. The three parameters that are grouped are equivalent up to constant factors. Parameters marked with $\star$ indicate the ones considered in this part.

**Proposition 3.5.** *Let $G$ be a graph and let $Y$ be a minimal blocking set of $G$.*

*(i) The minimal blocking set $Y$ contains no vertex that is contained in every optimum vertex cover of $G$.*

*(ii) If $Y$ contains a vertex that is not contained in any optimum vertex cover of $G$ then $|Y| = 1$. More precisely, the minimal blocking set $Y$ contains exactly one of these vertices.*

*(iii) The set $Y$ is contained in only one connected component of $G$.*

*(iv)* $\text{VC}(G - Y) + |Y| = \text{VC}(G) + 1$

**Proof.** To prove item (i) we assume for the sake of contradiction that there exists a vertex $y \in Y$ that is contained in every optimum vertex cover of $G$. Since $Y$ is a minimal blocking set of $G$ it holds that there exists an optimum vertex cover $X$ of $G$ that contains the set $Y \setminus \{y\}$. But, the vertex $y$ is contained in every optimum vertex cover of $G$, hence $Y \subseteq X$. This contradicts the assumption that $Y$ is a (minimal) blocking set of $G$ and proves that $Y$ contains no vertex that is contained in every optimum vertex cover of $G$.

Let $v$ be a vertex that is not contained in any optimum vertex cover of $G$. This implies that $\{v\}$ is a blocking set of $G$ (by definition). Thus, item (ii) holds.

Next, we prove item (iii). Assume for contradiction that the minimal blocking set $Y$ is contained in at least two connected components of $G$. Let $G'$ be a connected component of $G$ that contains at least one vertex of $Y$, and let $Y'$ be the set of vertices in $Y$ that are also contained in $G'$, i.e., $Y' = Y \cap V(G') \neq \emptyset$. Let $y \in Y'$ be an arbitrary vertex of $Y'$. Since $Y$ is a minimal blocking set of $G$ it holds that there exists an optimum vertex cover $X$ of $G$ with $Y \setminus \{y\} \subseteq X$. Thus, for every connected component $\hat{G}$ of $G$, except $G'$, there exists an optimum vertex cover $\hat{X}$ that contains the vertex set $Y \cap V(\hat{G})$. Observe that this holds also for the connected component $G'$ by choosing a vertex $\hat{y} \in Y \setminus Y'$ and considering an optimum vertex cover of $G$ that contains $Y \setminus \{\hat{y}\}$. But, this implies that there exists an optimum vertex cover of $G$ that contains $Y$ which contradicts the assumption that $Y$ is a blocking set of $G$. Hence, the minimal blocking set $Y$ is contained in at most one connected component of $G$.

Now, we prove item (iv). Since $Y$ is a blocking set of $G$ it holds by definition that $\text{VC}(G) < \text{VC}(G-Y)+|Y|$. For the other direction, let $y \in Y$ be an arbitrary vertex of $Y$. There exists an optimum vertex cover $X$ of $G$ that contains the set $Y \setminus \{y\}$ because $Y$ is a minimal blocking set of $G$. Since $X$ is a vertex cover of $G$ it follows that $X \setminus Y$ is a vertex cover of $G-Y$. Hence, $\text{VC}(G-Y) \leq |X \setminus Y| = |X| - |Y \setminus \{y\}| = \text{VC}(G) - (|Y|-1)$ which implies that $\text{VC}(G - Y) + |Y| = \text{VC}(G) + 1$. ∎

The following lemma shows that deleting a vertex set $Z$ from a graph $G$, that is contained in at least one optimum vertex cover of $G$, cannot increase the size of minimal blocking sets. Furthermore, it indicates that there is a strong relation between minimal blocking sets in $G$ and $G - Z$.

**Lemma 3.6.** *Let $G$ be a graph and let $Z \subseteq V(G)$ be a set of vertices such that there exists an optimum vertex cover $X$ of $G$ with $Z \subseteq X$.*

*(i)* $\textsc{vc}(G - Z) + |Z| = \textsc{vc}(G)$

*(ii) If $Y$ is a blocking set of $G$ then $Y \setminus Z$ is a blocking set of $G - Z$.*

*(iii) It holds that $\beta(G - Z) \leq \beta(G)$, more precisely, if $Y'$ is a minimal blocking set of $G - Z$ then there exists a (possibly empty) set $Z' \subseteq Z$ such that $Z' \cup Y'$ is a minimal blocking set of $G$.*

**Proof.** Obviously, item (i) holds, because every optimum vertex cover of $G-Z$ together with $Z$ is a vertex cover of $G$, and because there exists an optimum vertex cover of $G$ that contains $Z$.

Next, we prove item (ii) by contradiction. Let $Y$ be a blocking set of $G$ and assume that $Y \setminus Z$ is not a blocking set of $G - Z$. This implies that there exists an optimum vertex cover $X$ of $G - Z$ that contains the set $Y \setminus Z$. However, $X \cup Z$ is an optimum vertex cover of $G$ that contains $Y$ because $\textsc{vc}(G - Z) + |Z| = \textsc{vc}(G)$. This contradicts the assumption that $Y$ is a blocking set of $G$ and shows that the set $Y \setminus Z$ is a blocking set of $G - Z$.

Finally, we prove item (iii). Let $Y'$ be a minimal blocking set of $G - Z$. First, we show that $Y' \cup Z$ is a blocking set of $G$. Assume for contradiction that $Y' \cup Z$ is not a blocking set of $G$. Hence, there exists an optimum vertex cover $X$ of $G$ that contains $Y' \cup Z$. Since $\textsc{vc}(G-Z)+|Z| = \textsc{vc}(G)$ it holds that $X \setminus Z$ is an optimum vertex cover of $G - Z$ that contains $Y'$. This contradicts the assumption that $Y'$ is a blocking set of $G - Z$, and proves that $Y' \cup Z$ is a blocking set of $G$. To conclude the proof we will show that every minimal blocking set $Y \subseteq Y' \cup Z$ of $G$ contains the set $Y'$. Let $Y \subseteq Y' \cup Z$ be a minimal blocking set of $G$. It follows from item (ii) that $Y \setminus Z \subseteq Y'$ is a blocking set of $G - Z$. Since $Y'$ is a minimal blocking set of $G - Z$ it must hold that $Y \setminus Z = Y'$; thus, it holds that $Y' \subseteq Y$. ■

## 3.4. Comparing Above Lower Bound Parameters to $\mathcal{C}$-Modulators

In this section we want to point out the relation between the value of above lower bound parameters and the size of their corresponding $\mathcal{C}$-modulator, i.e., the class $\mathcal{C}$ of graphs where the parameter value is zero. First, we relate the above lower bound parameter $\textsc{vc} - \textsc{mm}$ with the size of a $\mathcal{C}$-modulator where $\mathcal{C}$ is the class of graphs where the size of a minimum vertex cover is equal to the size of a maximum matching. Afterwards we do the same for the above lower bound parameters $\textsc{vc} - \textsc{lp}$ and $\textsc{vc} - (2\textsc{lp} - \textsc{mm})$ and the size of their corresponding $\mathcal{C}$-modulators, i.e., the class of graphs where $\textsc{vc} = \textsc{lp}$ or $\textsc{vc} = 2\textsc{lp} - \textsc{mm}$, respectively. We will show that in all three cases there exists a $\mathcal{C}$-modulator that has at most twice the size of the corresponding above lower bound parameter.

**Lemma 3.7 (see also [MRS+11, Lemma 12]).** *Let $G$ be a graph, let $\ell = \textsc{vc}(G) - \textsc{mm}(G)$, and let $X^* \subseteq V(G)$ be an optimum $\mathcal{C}$-modulator, where $\mathcal{C}$ is the class of graphs where the size of a maximum matching is equal to the size of a minimum vertex cover. It holds that $\ell \leq |X^*| \leq 2\ell$.*

**Proof.** First, we show that $|X^*| \leq 2\ell$ by constructing a $\mathcal{C}$-modulator $X$ of size at most $2\ell$. Let $S$ be an optimum vertex cover of $G$ and let $M$ be a maximum matching of $G$. We define $X = (S \setminus V(M)) \cup \{v \in S \mid \exists w \in S \colon \{v, w\} \in M\}$ as the union of all vertices in $S$ that are not an endpoint of any matching edge of $M$ and all vertices in $S$ that are matched to a vertex in $S$ via $M$. The size of $X$ is

$$\begin{aligned}|X| &= |S \setminus V(M)| + |\{v \in S \mid \exists w \in S \colon \{v, w\} \in M\}| \\ &= (|S| - |M| - |\{e \in M \mid e \subseteq S\}|) + (2 \cdot |\{e \in M \mid e \subseteq S\}|) \\ &= \textsc{vc}(G) - \textsc{mm}(G) + |\{e \in M \mid e \subseteq S\}| \\ &\leq 2\ell,\end{aligned}$$

where the last inequality holds because at most $\ell$ matching edges can have both endpoints in $S$. Observe, that $M' = M \setminus \{e \in M \mid e \subseteq S\}$ is a perfect matching of $G - X$, and that $S \setminus X$ is an optimum vertex cover of $G - X$ that contains exactly one endpoint of every matching edge in $M'$. This implies that $\textsc{vc}(G - X) = \textsc{mm}(G - X)$ and shows that $X$ is a $\mathcal{C}$-modulator of $G$; hence $|X^*| \leq |X| \leq 2\ell$.

We showed that $|X^*| \leq 2\ell$. It remains to show that $\ell \leq |X^*|$. Obviously, it holds that $\textsc{mm}(G - X^*) \leq \textsc{mm}(G)$ and that $\textsc{vc}(G) \leq \textsc{vc}(G - X^*) + |X^*|$. Since $X^*$ is a $\mathcal{C}$-modulator it follows that $\textsc{vc}(G - X^*) = \textsc{mm}(G - X^*)$. Overall, this implies that

$$\ell = \textsc{vc}(G) - \textsc{mm}(G) \leq \textsc{vc}(G - X^*) + |X^*| - \textsc{mm}(G - X^*) = |X^*|.$$

This concludes the proof. ■

Next, we prove the relation between the value $\ell = \textsc{vc}(G) - \textsc{lp}(G)$ and the size of a $\mathcal{C}_{\textsc{lp}}$-modulator in $G$, where $\mathcal{C}_{\textsc{lp}}$ is the class of graphs where the size of an optimum vertex cover is equal to the size of an optimum LP solution.

**Lemma 3.8.** *Let $G$ be a graph, let $\ell = \textsc{vc}(G) - \textsc{lp}(G)$, and let $X^* \subseteq V(G)$ be an optimum $\mathcal{C}_{\textsc{lp}}$-modulator. It holds that $\ell \leq |X^*| \leq 2\ell$.*

**Proof.** As in the previous proof, we will show that there exists a $\mathcal{C}_{\textsc{lp}}$-modulator $X$ of size at most $2\ell$. Let $x \in \{0, \frac{1}{2}, 1\}^{|V(G)|}$ be an optimum half-integral solution to LP$(G)$. Due to a result of Nemhauser and Trotter [NT75] (Theorem 2.4) there exists an optimum vertex cover $S$ of $G$ with $V_1 \subseteq S \subseteq V_{\frac{1}{2}} \cup V_1$. Recall, $V_i = \{v \in V(G) \mid x_v = i\}$ for $i \in \{0, \frac{1}{2}, 1\}$. Consider the bipartite graph $H$ where one part is the set $V_{\frac{1}{2}} \cap S$ and the other part is the set $V_{\frac{1}{2}} \setminus S$, and where there is an edge between $y \in V_{\frac{1}{2}} \cap S$ and $z \in V_{\frac{1}{2}} \setminus S$ if and only if $\{y, z\} \in E(G)$. Let $M$ be a maximum matching in the bipartite graph $H$. It holds that the matching $M$ saturates the set $V_{\frac{1}{2}} \setminus S$: Otherwise, it follows from Theorem 2.2 (Hopcroft Karp [HK73]) that there exists a set $Z \subseteq V_{\frac{1}{2}} \setminus S$ such that $|N_H(Z)| < |Z|$. But, this implies that $x$ is not an optimum half-integral solution to LP($G$) because $x' \in \{0, \frac{1}{2}, 1\}^{|V(G)|}$, with $x'_v = 1$ for all $v \in V_1 \cup N_H(Z)$, $x'_v = \frac{1}{2}$ for all $v \in V_{\frac{1}{2}} \setminus N_H[Z]$, and $x'_v = 0$ for all $v \in V_0 \cup Z$, is also a feasible solution to LP($G$) with $w(x') < w(x)$. Note that $x'$ is a feasible solution to LP($G$) because $Z \cap S = \emptyset$ which implies that $Z$ is an independent set in $G$, and because $Z \subseteq V_{\frac{1}{2}}$ which implies that $N_G(Z) \subseteq V_{\frac{1}{2}} \cup V_1$. Thus, the matching $M$ saturates $V_{\frac{1}{2}} \setminus S$.

Let $X = (V_{\frac{1}{2}} \cap S) \setminus V(M)$ be the set of vertices in $V_{\frac{1}{2}} \cap S$ that are not an endpoint of a matching edge of $M$. We will show that $X$ is a $\mathcal{C}_{\text{LP}}$-modulator of size $2\ell$. By the choice of $S$, it holds that $\text{vc}(G) = |S| = |V_{\frac{1}{2}} \cap S| + |V_1|$ and that

$$\text{lp}(G) = |V_1| + \frac{1}{2}|V_{\frac{1}{2}}| = |V_1| + \frac{1}{2}|V_{\frac{1}{2}} \cap S| + \frac{1}{2}|V_{\frac{1}{2}} \setminus S|.$$

Since $\ell = \text{vc}(G) - \text{lp}(G)$ we obtain that

$$\begin{aligned}\ell = \text{vc}(G) - \text{lp}(G) &= |V_{\frac{1}{2}} \cap S| + |V_1| - \left(|V_1| + \frac{1}{2}|V_{\frac{1}{2}} \cap S| + \frac{1}{2}|V_{\frac{1}{2}} \setminus S|\right)\\ &= \frac{1}{2}|V_{\frac{1}{2}} \cap S| - \frac{1}{2}|V_{\frac{1}{2}} \setminus S| = \frac{1}{2}|(V_{\frac{1}{2}} \cap S) \setminus V(M)|\\ &= \frac{1}{2}|X|.\end{aligned}$$

The second to last equality holds because the matching $M$ in $H$ saturates $V_{\frac{1}{2}} \setminus S$. It follows that $|X| = 2\ell$. Observe that the graph $G - X$ has a matching of size $|V_1| + |M|$ and that $S \setminus X$ is a vertex cover of $G - X$ of size $|V_1| + |M|$. This implies that $\text{vc}(G - X) = \text{lp}(G - X)$ and shows that $X$ is a $\mathcal{C}_{\text{LP}}$-modulator of size $2\ell$.

Analogous to the proof of Lemma 3.7 it follows that $\ell \leq |X^*|$ because $\text{vc}(G) \leq \text{vc}(G - X^*) + |X^*|$, because $\text{lp}(G - X^*) \leq \text{lp}(G)$, and because $X^*$ is a $\mathcal{C}_{\text{LP}}$-modulator which implies that $\text{vc}(G - X^*) = \text{lp}(G - X^*)$. This concludes the proof. ■

Recall that we cannot bound the parameter value $\ell = \text{vc}(G) - \text{lp}(G)$ with the size of a modulator to a $d$-quasi-integral graph, because the graph $G$ that is the disjoint union of $c$ cliques $K_4$ is a 1-quasi integral graph, where $\text{vc}(G) - \text{lp}(G) = c$. However, it follows from Lemma 3.8 that there exists a modulator to a $d$-quasi-integral graph that contains at most $2 \cdot \ell$ many vertices.

To bound the size of a set $X$ such that $\text{vc}(G - X) = 2\text{lp}(G - X) - \text{mm}(G - X)$ when $\text{vc}(G) = 2\text{lp}(G) - \text{mm}(G) + \ell$ we use the structure of the Gallai-Edmonds decomposition of graph $G$ (see Theorem 2.3). First, we show that given an optimum vertex cover of $G$, there exists a maximum matching in $G$ that helps us to figure out where we spend the $\ell = \text{vc}(G) - (2\text{lp}(G) - \text{mm}(G))$ additional vertices.

**Lemma 3.9.** *Let $G$ be a graph, let $\ell = \text{VC}(G) - (2\text{LP}(G) - \text{MM}(G))$, let $x \in \{0, \frac{1}{2}, 1\}^{|V(G)|}$ be an optimum half-integral solution to* LP$(G)$*, and let $S$ be an optimum vertex cover of $G$ with $V_1 \subseteq S \subseteq V_{\frac{1}{2}} \cup V_1$. Let $(D, A, B)$ be the Gallai-Edmonds decomposition of $G[V_{\frac{1}{2}}]$, and let $D_S$ be the set of singleton components*[5] *of $G[D]$.*

*(i) For all sets $D'_S \subseteq D_S$ it holds that $|N_G(D'_S) \cap A| \geq |D'_S|$.*

*(ii) There exists a maximum matching $M$ in $G$ such that every $M$ exposed vertex is either contained in the optimum vertex cover $S$ or contained in the vertex set $V_0$.*

*(iii) Let $M$ be a maximum matching that fulfills the conditions of item (ii). It holds that $\ell = \{e \in M \mid e \subseteq S\}$.*

**Proof.** First, we show that for each set $D'_S \subseteq D_S$ it holds that $|N_G(D'_S) \cap A| \geq |D'_S|$. Note that $N_G(D'_S) \subseteq A \cup V_1$ for all $D'_S \subseteq D_S$. Assume for the sake of contradiction that there exists a set $D'_S \subseteq D_S$ such that $|N_G(D'_S) \cap A| < |D'_S|$. Let $D'_S$ be such an inclusion-wise minimal set. This implies that $x' \in \{0, \frac{1}{2}, 1\}^{|V(G)|}$ with $x'_v = 1$ for all $v \in V_1 \cup (N_G(D'_S) \cap A)$, $x'_v = \frac{1}{2}$ for all $v \in V_{\frac{1}{2}} \setminus (N_G[D'_S])$, and $x'(v) = 0$ for all $v \in V_0 \cup D'_S$ is also a feasible solution to LP$(G)$ because every neighbor of $D'_S$ is either contained in $A$ or $V_1$. Furthermore, it holds that $w(x') < w(x)$ because $|N_G(D'_S) \cap A| < |D'_S|$. This contradicts the assumption that $x$ is an optimum half-integral solution to LP$(G)$ and proves item (i).

Overall, this implies that $|D_S| \leq |A|$ and that there exists a matching in $G[A \cup D_S]$ that saturates $D_S$ (Hall's Theorem, Theorem 2.1). Let $M'$ be a matching in $G[A \cup D_S]$ that saturates $D_S$. We construct a matching that fulfills item (ii) using matching $M'$. Since $x$ is an optimum half-integral solution to LP$(G)$ it holds that there exists a maximum matching between $V_0$ and $V_1$ that saturates $V_1$. Let $M_1$ be a maximum matching between $V_0$ and $V_1$ that saturates $V_1$. Furthermore, we can assume that a maximum matching of $G$ always contains the matching $M_1$ because we can replace for every maximum matching $M$ of $G$ the edges that are incident with $V_1$ by the edges in $M_1$ to obtain a maximum matching of $G$ that contains $M_1$. Thus, it is enough to construct a maximum matching $\hat{M}$ of $G[V_{\frac{1}{2}}]$ that fulfills the requirements of item (ii).

We will show that there exists a maximum matching in $G$ that saturates all vertices of the set $D_S$. Let $M_2$ be a maximum matching between $A$ and $D$ that contains at most one vertex of each connected component of $G[D]$ and with $D_S \setminus V(M_2)$ minimal. Note that $M_2$ saturates $A$ because every maximum matching of $G$ matches the vertices of $A$ to different connected components of $G[D]$ (property of the Gallai-Edmonds decomposition $(D, A, B)$ of $G[V_{\frac{1}{2}}]$). Assume for contradiction that there exists a vertex $v \in D_S \setminus V(M_2)$. Let $D' \subseteq D$ be the set of vertices in $D$ that are reachable from $v$ by an $M_2$-alternating path. Note that every vertex in $D'$ is endpoint of an edge in $M_2$. If $D' \setminus D_S \neq \emptyset$ then there exists an $M_2$-alternating path from $v$ to a vertex in $D \setminus D_S$. Augmenting along this path

[5] Recall, a singleton component is a connected component of size one.

leads to a maximum matching $M_2'$ between $A$ and $D$ with $D_S \setminus V(M_2') \subsetneq D_S \setminus V(M_2)$ which contradicts the assumption that $D_S \setminus V(M_2)$ is minimal. Thus, it holds that $D' \subseteq D_S$. But, this implies that $|N_G(D') \cap A| < |D'|$ which is not possible due to item (i). Therefore, we have shown that $M_2$ saturates $D_S$.

Finally, we are able to construct the matching $\hat{M}$. We add a perfect matching of $G[B]$ as well as the matching $M_2$ to $\hat{M}$. For every connected component $C$ of $G[D \setminus D_S]$ we pick a vertex $v \in V(C) \cap S$, which exists because $|V(C)| \geq 3$, when $V(C) \cap V(M_2) = \emptyset$ and the one vertex $v \in V(C) \cap V(M_2)$ when $V(C) \cap V(M_2) \neq \emptyset$ and add a perfect matching of $C - v$ to $\hat{M}$. The existence of such a matching follows from the fact that $C$ is factor-critical. It follows from the Gallai-Edmonds decomposition of $G[V_{\frac{1}{2}}]$ that $\hat{M}$ is maximum matching of $G[V_{\frac{1}{2}}]$. Furthermore, $\hat{M}$ fulfills the requirements of item (ii) for $G - V_1 - V_0$ because every vertex that is not saturated by $\hat{M}$ is contained in a factor-critical component of $G[D \setminus D_S]$ and by construction it holds that this vertex is in $S$. Now, $M = \hat{M} \cup M_1$ fulfills the requirements of item (ii).

It remains to prove item (iii). Therefore, we consider the values for $\textsc{vc}(G)$, $\textsc{lp}(G)$, and $\textsc{mm}(G)$. It holds that $\textsc{vc}(G) = |S| = |V_1| + |S \cap V_{\frac{1}{2}}|$. Furthermore, we know that $2\textsc{lp}(G) = 2|V_1| + |V_{\frac{1}{2}}| = 2|V_1| + |V_{\frac{1}{2}} \cap S| + |V_{\frac{1}{2}} \setminus S|$. Let $M$ be a maximum matching of $G$ that fulfills item (ii) and contains a maximum matching between $V_0$ and $V_1$. As above, let $M_1 \subseteq M$ be the matching between $V_0$ and $V_1$ of size $|V_1|$ and let $\hat{M} \subseteq M$ be the matching of $G[V_{\frac{1}{2}}]$ that contains every vertex of $V_{\frac{1}{2}}$ that is not contained in $S$; hence $\textsc{mm}(G) = |M| = |M_1| + |\hat{M}| = |V_1| + |\hat{M}|$. Overall, we obtain:

$$
\begin{aligned}
\ell &= \textsc{vc}(G) - (2\textsc{lp}(G) - \textsc{mm}(G)) \\
&= |V_1| + |S \cap V_{\frac{1}{2}}| - \Big(2|V_1| + |V_{\frac{1}{2}} \cap S| + |V_{\frac{1}{2}} \setminus S| - \big(|V_1| + |\hat{M}|\big)\Big) \\
&= |V_1| + |S \cap V_{\frac{1}{2}}| - 2|V_1| - |V_{\frac{1}{2}} \cap S| - |V_{\frac{1}{2}} \setminus S| + |V_1| + |\hat{M}| \\
&= -|V_{\frac{1}{2}} \setminus S| + |\hat{M}| = |\{e \in M \mid e \subseteq S\}|,
\end{aligned}
$$

where the last equality holds because every vertex in $V_{\frac{1}{2}}$ that is not contained in $S$ is endpoint of a matching edge $e \in \hat{M} \subseteq M$, and because every matching edge $e \in M_1$ has only one endpoint in $S$. This concludes the proof of the lemma. ∎

**Lemma 3.10.** *Let $G$ be a graph, let $\ell = \textsc{vc}(G) - (2\textsc{lp}(G) - \textsc{mm}(G))$, and let $X^*$ be an optimum $\mathcal{C}$-modulator, where $\mathcal{C}$ is the class of graphs where the size of a minimum vertex cover is equal to two times the size of an optimum* LP *solution minus the size of a maximum matching, i.e., $\mathcal{C} = \{G \mid \textsc{vc}(G) = 2\textsc{lp}(G) - \textsc{mm}(G)\}$. It holds that $\frac{1}{2}\ell \leq |X^*| \leq 2\ell$.*

**Proof.** Let $x \in \{0, \frac{1}{2}, 1\}^{|V(G)|}$ be an optimum half-integral solution to LP$(G)$, and let $S$ be an optimum vertex cover of $G$ with $V_1 \subseteq S \subseteq V_{\frac{1}{2}} \cup V_1$ which exists due to a result of Nemhauser and Trotter [NT75] (Theorem 2.4). Let $(D, A, B)$ be the unique

Gallai-Edmonds decomposition of $G[V_{\frac{1}{2}}]$. Let $M$ be a maximum matching of $G$ that fulfills the properties of item (ii) of Lemma 3.9. Let $X = \{y \in S \mid \exists z \in S\colon \{y,z\} \in M\}$ be the set of vertices in $S$ that are matched to vertices in $S$ via $M$. We will show that $X$ is a $\mathcal{C}$-modulator of size $2\ell$ which implies that $|X^*| \leq 2\ell$.

Observe that $|X| = 2\ell$, because $|X| = 2 \cdot |\{e \in M \mid e \subseteq S\}| = 2\ell$ (Lemma 3.9 item (iii)). Now, we consider the graph $G - X$. It holds that $\textsc{vc}(G-X) + |X| = \textsc{vc}(G)$ because $S$ is an optimum vertex cover of $G$ that contains the set $X$. Since $X$ is the union of all matching edges of $M$ where both endpoints are contained in $S$ it holds that $\textsc{mm}(G-X) = \textsc{mm}(G) - \frac{1}{2}|X|$. Furthermore, it holds that $\textsc{lp}(G-X) \leq \textsc{lp}(G) - \frac{1}{2}|X|$ because $X \subseteq V_{\frac{1}{2}}$. Next, we show that $\textsc{lp}(G-X) \geq \textsc{lp}(G) - \frac{1}{2}|X|$ to prove that $\textsc{lp}(G-X) = \textsc{lp}(G) - \frac{1}{2}|X|$.

Assume for contradiction that $\textsc{lp}(G-X) < \textsc{lp}(G) - \frac{1}{2}|X|$. Let $x' \in \{0, \frac{1}{2}, 1\}^{|V(G-X)|}$ be an optimum half-integral solution to LP$(G-X)$. We can assume that $V_1^x \subseteq V_1^{x'}$ and that $V_0^x \subseteq V_0^{x'}$ because $X \subseteq V_{\frac{1}{2}}$. Since $w(x') + \frac{1}{2}|X| = \textsc{lp}(G-X) + \frac{1}{2}|X| < \textsc{lp}(G) = w(x)$ it holds that $|V_{\frac{1}{2}}^x \cap V_1^{x'}| < |V_{\frac{1}{2}}^x \cap V_0^{x'}|$. This implies that there exists a vertex $v \in V_{\frac{1}{2}}^x \cap V_0^{x'}$ that is not contained in $M$ because $N_{G-X}(V_{\frac{1}{2}}^x \cap V_0^{x'}) \subseteq V_1^{x'}$, because $V_1^{x'} \cap V_1^x = V_1^x$ is matched to $V_0^{x'} \cap V_0^x = V_0^x$, and because $M$ contains a perfect matching of $G[X]$. Thus, the vertex $v$ is contained in $S$ because every vertex that is exposed by $M$ is contained in $S$. Now, $S \setminus (V_{\frac{1}{2}}^x \cap V_0^{x'}) \cup (V_{\frac{1}{2}}^x \cap V_1^{x'})$ is an optimum vertex cover of $G$ of size at most $|S| - 1$. This contradicts the assumption that $S$ is an optimum solution to $G$ and concludes the proof that $\textsc{lp}(G-X) \geq \textsc{lp}(G) - \frac{1}{2}|X|$.

Combining the above equations we obtain that

$$\begin{aligned}
&\textsc{vc}(G-X) - (2\textsc{lp}(G-X) - \textsc{mm}(G-X)) \\
&\quad = \textsc{vc}(G) - |X| - (2\textsc{lp}(G) - |X| - (\textsc{mm}(G) - \frac{1}{2}|X|)) \\
&\quad = \ell - \frac{1}{2}|X| = 0
\end{aligned}$$

because $|X| = 2\ell$. This shows that $X$ is a $\mathcal{C}$-modulator of size at most $2\ell$ and proves that $|X^*| \leq 2\ell$.

It remains to show that $\ell \leq 2|X^*|$. We know that $\textsc{vc}(G) \leq \textsc{vc}(G-X^*) + |X^*|$, that $\textsc{mm}(G) \leq \textsc{mm}(G-X^*) + |X^*|$, and that $\textsc{lp}(G-X^*) \leq \textsc{lp}(G)$. Combining these inequalities we obtain that

$$\begin{aligned}
\ell &= \textsc{vc}(G) - (2\textsc{lp}(G) - \textsc{mm}(G)) \\
&\leq \textsc{vc}(G-X^*) + |X^*| - (2\textsc{lp}(G-X^*) - (\textsc{mm}(G-X^*) + |X^*|)) \\
&= 2|X^*|.
\end{aligned}$$

This concludes the proof. ■

**Overview of this part.** Motivated by the great variety of positive and negative results for kernelization for Vertex Cover subject to different parameters and graph classes, we seek to unify and generalize them using blocking sets, which have played implicit and explicit roles in many results. More precisely, most lower bounds use that the graph class has unbounded minimal blocking set size. Furthermore, many kernels use that the minimal blocking set size is bounded. We show in Chapter 4 that this is no coincidence. More precisely, we show that in the most-studied setting, parameterized by the size of a $\mathcal{C}$-modulator, bounded minimal blocking set size is necessary but not sufficient to get a polynomial kernelization. Under mild technical assumptions, bounded minimal blocking set size allows an essentially tight efficient reduction in the number of connected components.

In Chapter 5 we first determine the exact maximum size of minimal blocking sets for $d$-quasi-forests, $d$-quasi-bipartite graphs, and $d$-quasi-integral graphs. We then extend these results for graphs of bounded elimination distance to any hereditary class $\mathcal{C}$, where $\beta_{\mathcal{C}}$ is bounded, including the case of graphs of bounded treedepth. Here, we also determine the exact maximum size of minimal blocking sets. We get similar but not tight bounds for certain non-hereditary classes $\mathcal{C}$, including the class $\mathcal{C}_{\text{LP}}$ of graphs where integral and fractional vertex cover size coincide.

Afterwards, in Chapter 6 we use the above results to obtain polynomial kernels for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator or $(\mathcal{C}, d)$-modulator for certain graph classes $\mathcal{C}$. This allow us, for example, to derive polynomial kernels for Vertex Cover parameterized by the size of a $(\mathcal{C}, d)$-modulator where $\mathcal{C}$ is the class of forests, bipartite graphs, or graphs where the optimum vertex cover equals the LP solution. Finally, we conclude this part in Chapter 7.

Throughout this part we will assume that all graph classes $\mathcal{C}$ are robust. Observe that this does not influence the blocking set size. For an arbitrary graph class $\mathcal{C}$ we can define the "robust closer" of $\mathcal{C}$, i.e., a graph class $\mathcal{C}^*$ that contains all graphs $G$ where each connected component $H$ of $G$ is a connected component of a graph in $\mathcal{C}$. Obviously, graph class $\mathcal{C}^*$ is robust and it holds that $\mathcal{C} \subseteq \mathcal{C}^*$. Now, every $\mathcal{C}$-modulator or $(\mathcal{C}, d)$-modulator is also a $\mathcal{C}^*$-modulator or $(\mathcal{C}^*, d)$-modulator, respectively. Thus, $\beta_{\mathcal{C}} \leq \beta_{\mathcal{C}^*}$ and $\beta_{\mathcal{C}}(d) \leq \beta_{\mathcal{C}^*}(d)$. Note that all graph classes $\mathcal{C}$, except $\mathcal{C}$ being the class of single cliques, that were considered so far for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator, are robust.

This part mostly follows joint work with Stefan Kratsch and Astrid Pieterse [HKP20]. The results of Section 5.2 follow joint work with Stefan Kratsch [HK17].

# CHAPTER 4
# RELATION BETWEEN BLOCKING SETS AND KERNELS

## 4.1. Introduction

Most known polynomial kernelizations for Vertex Cover are for parameterization by the vertex deletion distance to some fixed hereditary graph class $\mathcal{C}$ that is also robust, e.g., for $\mathcal{C}$ being the class of forests [JB13], graphs of maximum degree one or two [MRS18], pseudoforests [FS16], bipartite graphs [KW12], cluster graphs of bounded clique size [MRS18], or graphs of bounded treedepth [BS19]. As mentioned before, the input is of form $(G, k, X)$, asking whether $G$ has a vertex cover of size at most $k$, where $X \subseteq V$ such that $G - X \in \mathcal{C}$; the size $\ell = |X|$ of the $\mathcal{C}$-modulator $X$ is the parameter. Blocking sets have been implicitly or explicitly used for most of these results and we point out that all the mentioned classes have bounded minimal blocking set size.

As our first result, we show that this is not a coincidence: If $\mathcal{C}$ is closed under disjoint union (or, more strongly, if $\mathcal{C}$ is robust) then bounded size of minimal blocking sets in graphs of $\mathcal{C}$ is necessary for a polynomial kernel to exist (Section 4.2). Moreover, the maximum size of minimal blocking sets in $\mathcal{C}$ yields a lower bound for the possible kernel size.

**Theorem 4.1.** *Let $\mathcal{C}$ be a graph class that is closed under disjoint union. If $\mathcal{C}$ contains any graph with a minimal blocking set of size $d$ then* Vertex Cover *parameterized by the size of a $\mathcal{C}$-modulator $X$ does not have a kernelization of size $\mathcal{O}(|X|^{d-\varepsilon})$ for any $\varepsilon > 0$ unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ *and the polynomial hierarchy collapses.*

To the best of our knowledge, this theorem captures all known kernel lower bounds for Vertex Cover parameterized by the deletion distance to any union-closed graph class $\mathcal{C}$, e.g., ruling out polynomial kernels for $\mathcal{C}$ being the class of mock-forests [FS16], outerplanar graphs [Jan13], or any class containing all cliques [BJK14]; and getting kernel size lower bounds for graphs of bounded treedepth [BS19] or cluster graphs of bounded clique size [MRS18]. To get lower bounds of this type, it now suffices to prove (or observe) that $\mathcal{C}$ has large or even unbounded minimal blocking set size.

It is natural to ask whether the converse holds, i.e., whether a bound on the minimal blocking set size directly implies the existence of a polynomial kernelization. Unfortunately, we show that this does not hold in a strong sense: There is a class $\mathcal{C}$ such that all graphs in $\mathcal{C}$ have minimal blocking sets of size one, but there is no polynomial kernelization (Section 4.3). More strongly, solving Vertex Cover on $\mathcal{C}$ is not in $\mathsf{RP} \supseteq \mathsf{P}$ unless $\mathsf{NP} = \mathsf{RP}$.

**Theorem 4.2.** *There exists a graph class $\mathcal{C}$ such that all graphs in $\mathcal{C}$ have minimal blocking set size one and such that* Vertex Cover *on $\mathcal{C}$ is not solvable in polynomial time (in fact, not in $\mathsf{RP}$), unless $\mathsf{NP} = \mathsf{RP}$.*

In light of this result, one could ask what further assumptions on $\mathcal{C}$, apart from the necessity of bounded minimal blocking set size, are required to allow for polynomial kernels. Clearly, polynomial-time solvability of Vertex Cover on the class $\mathcal{C}$ is necessary and (as we implicitly showed) not implied by $\mathcal{C}$ having bounded blocking set size. If, slightly stronger, we require that blocking sets in graphs of $\mathcal{C}$ can be efficiently recognized[1] then we show that there is an efficient algorithm that reduces the number of components of $G - X$ for any instance $(G, k, X)$ of Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator to $\mathcal{O}(|X|^d)$ (Section 4.4). This is a standard opening step for kernelization and can be followed up by shrinking and bounding the size of those components (unless bounds follow directly from $\mathcal{C}$ such as for cluster graphs with bounded clique size). Note that this requires that deletion of any component yields a graph in $\mathcal{C}$ (e.g., implied by $\mathcal{C}$ being robust), which here is covered already by $\mathcal{C}$ being hereditary.

**Theorem 4.3.** *Let $\mathcal{C}$ be any hereditary graph class with minimal blocking set size $d$ on which* Vertex Cover *can be solved in polynomial time. There is an efficient algorithm that given $(G, k, X)$ such that $G - X \in \mathcal{C}$ returns an equivalent instance $(G', k', X)$ such that $G' - X \in \mathcal{C}$ has at most $\mathcal{O}(|X|^d)$ connected components.*

We point out that the number $\mathcal{O}(|X|^d)$ of components is essentially tight (assuming $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$) because the lower bound underlying Theorem 4.1 creates instances where components have a constant number $c = c(d)$ of vertices. Reducing to $\mathcal{O}(|X|^{d-\varepsilon})$ components, for any $\varepsilon > 0$, would violate the kernel size lower bound.

## 4.2. Polynomial Kernel Implies a Bound on the Minimal Blocking Set Size

In this section we prove Theorem 4.1, showing that if $\mathcal{C}$ is a graph class where minimal blocking sets can have size $d$, then this gives a kernelization lower bound for Vertex

[1] This condition clearly holds for all hereditary classes $\mathcal{C}$ on which Vertex Cover can be solved in polynomial time: Given $G = (V, E)$ and $Y \subseteq V$ it suffices to compute solutions for $G$ and $G - Y$. Clearly, the set $Y$ is a blocking set if and only if $\mathrm{VC}(G) < \mathrm{VC}(G - Y) + |Y|$.

Cover when parameterized by the size of a $\mathcal{C}$-modulator. Thus, under the assumption that $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$, the theorem shows that having bounded blocking set size is necessary to obtain a polynomial kernel in the following sense. For a graph class $\mathcal{C}$ closed under disjoint union, for which Vertex Cover parameterized by a modulator to $\mathcal{C}$ admits a polynomial kernel of size $\mathcal{O}(k^d)$, it must hold that $\beta_{\mathcal{C}} \leq d$.

**Theorem 4.1.** *Let $\mathcal{C}$ be a graph class that is closed under disjoint union. If $\mathcal{C}$ contains any graph with a minimal blocking set of size $d$ then* Vertex Cover *parameterized by the size of a $\mathcal{C}$-modulator $X$ does not have a kernelization of size $\mathcal{O}(|X|^{d-\varepsilon})$ for any $\varepsilon > 0$ unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ *and the polynomial hierarchy collapses.*

**Proof.** For $d = 1$, observe that a kernel of size $\mathcal{O}(|X|^{1-\varepsilon})$ can be ruled out by the following argument. Suppose such a kernel exists for $\varepsilon > 0$. Let $G$ be an undirected graph, and let $k \in \mathbb{N}$. Let $H$ be an arbitrary constant-size graph in $\mathcal{C}$. Let $G'$ be the disjoint union of $H$ and $G$, implying that $G'$ has a $\mathcal{C}$-modulator of size $|V(G)|$, namely the set $V(G)$. Let $k' := k + \textsc{vc}(H)$. It is easy to observe that $G'$ has a vertex cover of size $k'$ if and only if $G$ has a vertex cover of size $k$. However, using the hypothetical kernel we can solve $(G', k', V(G))$ in polynomial time, by repeatedly applying the kernelization algorithm until we obtain a constant-size instance. This would imply that $\mathsf{P} = \mathsf{NP}$, implying $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

For $d \geq 2$, the lower bound is obtained by a linear-parameter transformation from $d$-Vertex Cover. An input to this problem consists of a $d$-uniform hypergraph[2] $G$ and an integer $k$. The problem is to decide whether $G$ has a vertex cover of size at most $k$. A vertex cover of a hypergraph is a set $X \subseteq V(G)$ such that for every edge $e \in E(G)$ we have $e \cap X \neq \emptyset$.

The lower bound will then follow from the fact that for $d \geq 2$, $d$-Vertex Cover parameterized by the number of vertices $n$ does not have a kernel of size $\mathcal{O}(n^{d-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [DvM14, Theorem 2].

Suppose, we are given an instance $(G, k)$ for $d$-Vertex Cover with vertex set $V(G) = \{x_1, x_2, \ldots, x_n\}$ and edge set $E(G) = \{e_1, e_2, \ldots, e_m\}$. We show how to construct an instance $(G', k', X)$ for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator with $|X| = n$. Refer to Figure 4.1 for a sketch of $G'$.

Start by adding a vertex $v_i$ for all $i \in [n]$ to $G'$ and let $X := \{v_i \mid i \in [n]\}$ be the union of these vertices. The set $X$ will be a $\mathcal{C}$-modulator of $G$. Let $H \in \mathcal{C}$ be a graph with a minimal blocking set of size $d$. For all $j \in [m]$, create a new copy of $H$ called $H_j$ and add it to $G'$. Choose a minimal blocking set $B_j$ of size $d$ in $H_j$ and enumerate the vertices as $B_j := \{b_1^j, \ldots, b_d^j\}$. Now, we connect the vertices of graphs $H_j$ to vertices in $X$, depending on the vertices contained in edge $e_j$. For each $j \in [m]$, for each $q \in [d]$, if the $q$'th vertex in edge $e_j$ equals $x_i$, connect vertex $b_q^j$ to vertex $v_i$. This concludes the construction of $G'$.

[2] A hypergraph $G = (V, E)$ consists of a finite set $V$ of vertices and a set $E$ of subsets of $V$, called (hyper)edges. A hypergraph is $d$-uniform, if every (hyper)edge consists of exactly $d$ vertices.

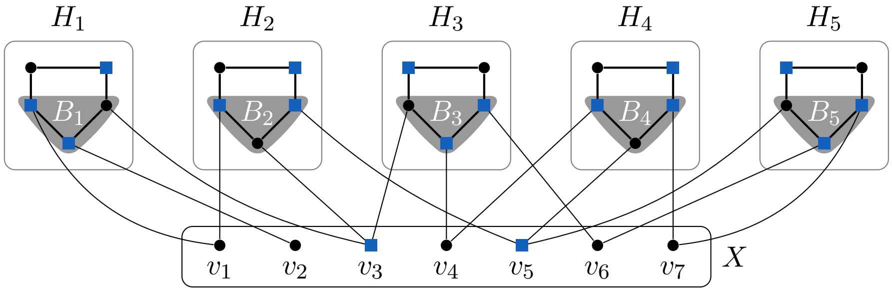



Figure 4.1.: The VERTEX COVER instance $G'$ with set $X$ obtained in the proof of Theorem 4.1, corresponding to instance $(G, 2)$ of 3-VERTEX COVER with $V(G) = \{x_i \mid i \in [7]\}$, and $E(G) = \{\{x_1, x_2, x_3\}, \{x_1, x_3, x_5\}, \{x_3, x_4, x_6\}, \{x_4, x_5, x_7\}, \{x_5, x_6, x_7\}\}$. Since $G$ has a vertex cover of size $k = 2$, $G'$ has a vertex cover of size $k' = 2 + 5 \cdot 3 = 17$, indicated with the blue rectangles.

Observe that $X$ is a $\mathcal{C}$-modulator of $G'$ since $G' - X$ consists of disjoint copies of $H \in \mathcal{C}$, and $\mathcal{C}$ is closed under disjoint union. Furthermore, $|X| = n = |V(G)|$ which is appropriately bounded for a linear-parameter transformation. Let $k' := m \cdot \text{VC}(H) + k$, we show that graph $G'$ has a vertex cover of size $k'$, if and only if the hypergraph $G$ has a vertex cover of size $k$.

($\Rightarrow$) Suppose $G$ has a vertex cover $S$ of size $k$, we show how to construct a vertex cover $S'$ of size $k'$ in $G'$. For $i \in [n]$, if $x_i \in S$ then let $v_i \in S'$. This completely defines $X \cap S'$. We show how to extend $S'$ to a vertex cover of the entire graph, using at most $\text{VC}(H)$ vertices from each copy of $H$.

For all $j \in [m]$, let $B'_j := \{b \in B_j \mid \exists x \in X \setminus S\colon \{b, x\} \in E(G')\}$. Observe that $B'_j \subsetneq B_j$, since there is at least one vertex $b \in B_j$ such that its unique neighbor in $X$ is contained in $S'$, as $S$ is a vertex cover of $G$ and the vertices in $N_{G'}(B_j) \cap X$ correspond to an edge in $G$. Add a minimum vertex cover of $H_j$ that contains $B'_j$ to $S'$. Since $B'_j \subsetneq B_j$ and $B_j$ is a minimal blocking set for $H_j$, it follows that this vertex cover has size $\text{VC}(H)$. Hereby, $S'$ has size $k'$ and it is easy to verify that $S'$ is indeed a vertex cover of $G'$.

($\Leftarrow$) Suppose $G'$ has a vertex cover $S'$ of size $k' = k + m \cdot \text{VC}(H)$. We start by showing that we can modify $S'$ such that $|S' \cap H_j| = \text{VC}(H)$ for all $j \in [m]$. Suppose that for some $j \in [m]$ it holds that $|S' \cap V(H_j)| > \text{VC}(H)$. Choose an arbitrary $b \in B_j$ and add the unique vertex in $N_{G'}(b) \cap X$ to $S'$. Replace $S' \cap V(H_j)$ by a vertex cover of $H_j$ of size $\text{VC}(H)$ that contains all vertices in $B_j \setminus \{b\}$. Observe that such a vertex cover exists since $B_j$ is a minimal blocking set. By construction, the resulting set is a vertex cover of $G'$ of size at most $k'$ with $|S' \cap V(H_j)| = \text{VC}(H)$ for all $j \in [m]$. As such, we from now on assume that $|S' \cap V(H_j)| = \text{VC}(H)$ for all $j \in [m]$.

Define $S := \{x_i \mid v_i \in S' \cap X, i \in [n]\}$ and observe that $|S| = k$. It remains to show that $S$ is a vertex cover of $G$. Suppose there is an edge $e_j \in E(G)$ such that

$e_j \cap S = \emptyset$. But then, for all $b \in B_j$ we have that there exists a vertex $x \in X$ such that $\{b, x\} \in E(G')$ and $x \notin S'$. Since $S'$ is a vertex cover of $G'$ that implies $S' \cap V(H_j)$ is a vertex cover of $H_j$ containing all vertices from $B_j$. This however contradicts the assumption that $|S' \cap V(H_j)| = \textsc{vc}(H)$ or that $B_j$ is a blocking set. Thus, $S$ is a vertex cover of $G'$ which concludes the proof. ∎

The above theorem shows that the existence of large minimal blocking sets allows us to prove kernelization lower bounds for Vertex Cover. It can be seen that many of the existing lower bound results for Vertex Cover when parameterized by the size of a $\mathcal{C}$-modulator for some graph class $\mathcal{C}$, relied on this same idea. Jansen [Jan13, Theorem 5.3] showed that Vertex Cover parameterized by a modulator to outerplanar graphs has no polynomial kernel, by a polynomial parameter transformation starting from $d$-CNF-SAT. The proof relies on the construction of an outerplanar clause gadget with blocking set size $d$, for any $d$. The same construction was used to obtain a lower bound for Vertex Cover parameterized by a modulator to constant treedepth [JP18, Theorem 2]. Furthermore, the result of Fomin and Strømme [FS16, Theorem 2], which states that Vertex Cover parameterized by the size of a modulator to a mock-forest is unlikely to have a polynomial kernel, uses that a mock-forest has unbounded minimal blocking set size. When choosing $\mathcal{C}$ as the graph class consisting of all cluster graphs in which each clique has size at most $d$, a lower bound was obtained by Majumdar et al. [MRS18]. They showed that the problem is unlikely to have a kernel of size $\mathcal{O}(n^{d-\varepsilon})$, again relying on the fact that a size-$d$ clique has minimal blocking set size $d$.

As such, Theorem 4.1 generalizes most existing lower bounds for Vertex Cover when parameterized by the size of a modulator to $\mathcal{C}$ for some graph class $\mathcal{C}$. We however mention that the polynomial kernel lower bound for Vertex Cover parameterized by the size of a modulator to a single clique [BJK14] does not fit our framework. While cliques have unbounded blocking set size, the class of single cliques is not closed under disjoint union. This introduces additional difficulties when constructing the type of transformation we give above. Indeed, the lower bound for Vertex Cover parameterized by the size of a modulator to a single clique is obtained by a cross-composition.

## 4.3. Bounded Minimal Blocking Set Size Is Not Sufficient

Now that it is clear, that proving that a graph class has bounded minimal blocking set size is essential towards obtaining a polynomial kernel for Vertex Cover parameterized by the size of a modulator to this graph class, one may wonder whether this condition is also sufficient. It turns out that it is *not*. Even worse, there exists a graph class $\mathcal{C}$ for which all minimal blocking sets have size 1, for which Vertex Cover is unlikely to be solvable in polynomial time. As such, Vertex Cover parameterized by the size of a modulator to $\mathcal{C}$ is unlikely to be fixed-parameter tractable, as Vertex Cover is not likely to be solvable in polynomial time when the parameter value is zero. Since a problem that is not fixed-parameter tractable does not admit any kernel at all, this proves our result. More precisely, we obtain the following theorem. (The graph class $\mathcal{C}$ obtained in the proof could be made robust by taking its closure under disjoint union and deletion of components without breaking the fact that a polynomial-time algorithm for Vertex Cover is unlikely.)

**Theorem 4.2.** *There exists a graph class $\mathcal{C}$ such that all graphs in $\mathcal{C}$ have minimal blocking set size one and such that* Vertex Cover *on $\mathcal{C}$ is not solvable in polynomial time (in fact, not in* RP*), unless* NP = RP*.*

**Proof.** It is known that the Unique-SAT problem cannot be solved in polynomial time unless NP = RP [VV86, Corollary 1.2]. An input to Unique-SAT is a CNF-formula $\mathcal{F}$ that has either exactly one satisfying solution or is unsatisfiable. The problem is to decide whether $\mathcal{F}$ is satisfiable. It can be shown that the same result also holds for Unique-3-SAT [Koz92, Example 26.7], where the input formula is further restricted to be in 3-CNF.

We show that the following polynomial-time reduction from Unique-3-SAT to Vertex Cover exists.

Claim 4.4. *There is a polynomial-time reduction from* Unique-3-SAT *to* Vertex Cover*, that given a formula $\mathcal{F}$, outputs an instance $(G, k)$ for* Vertex Cover *such that:*

- *If $\mathcal{F}$ has exactly one satisfying assignment, then $G$ has a unique minimum vertex cover of size $k$.*
- *If $\mathcal{F}$ is unsatisfiable, then $G$ has a unique minimum vertex cover of size $k + 1$.*

Proof. Instead of showing the above result for Vertex Cover, we will, for simplicity, reduce from Unique-3-SAT to Clique. We give a reduction such that if $\mathcal{F}$ has a unique satisfying assignment, then $G$ has a unique clique of size $\ell$, and otherwise $G$ has a unique clique of size $\ell - 1$. Since $S \subseteq V(G)$ is a clique in $G$ if and only if $V(G) \setminus S$ is a vertex cover in $\bar{G}$, the complement of $G$, the desired reduction for Vertex Cover follows from taking the complement of $G$ and letting $k := |V(G)| - \ell$.

Let a formula $\mathcal{F}$ be given. There is a parsimonious reduction from Unique-3-SAT to Clique (see for example [Koz92, Example 26.8]), meaning that there is a polynomial-time many-one reduction from Unique-3-SAT to Clique such that the number of size-$\ell$ cliques in $G$ corresponds to the number of satisfying assignments of $\mathcal{F}$. Use this reduction to obtain an instance $(G, \ell)$ for Clique. Clearly, $G$ has the property that if $\mathcal{F}$ has a unique satisfying assignment, then $G$ has a unique maximum clique. However, we also want to guarantee that $G$ has a unique (albeit smaller) maximum clique when $\mathcal{F}$ is unsatisfiable.

We now show how to obtain an instance $G'$ satisfying both these requirements. Initialize $G'$ as two copies of graph $G$, say $G_1$ and $G_2$. Label the vertices of $G_1$ and $G_2$ as $v_1^1, \ldots, v_n^1$ and $v_1^2, \ldots, v_n^2$ such that $G_1$ and $G_2$ are isomorphic by the function mapping $v_i^1$ to $v_i^2$ for all $i \in [n]$. For all $i \in [n]$, add the edge $\{v_i^1, v_i^2\}$ to $G'$. Furthermore, add the edges $\{\{v_i^1, v_j^2\} \mid \{v_i^1, v_j^1\} \in E(G_1)\}$ to $G'$. In this way, vertices $v_i^1$ and $v_i^2$ have the same closed neighborhood in $G'$, for any $i \in [n]$. We conclude the construction of $G'$ by adding a new set of vertices $Z$ of size $2\ell - 1$ to the graph, and letting $Z$ be a clique. We show that the conditions of the lemma statement hold for $(G', \ell')$ where we let $\ell' := 2\ell$.

Suppose $\mathcal{F}$ has exactly one satisfying assignment. By this definition, there is a size-$\ell$ clique in $G_1$, let this clique be $K_1$. Define $K_2 := \{v_i^2 \mid v_i^1 \in K_1, i \in [n]\}$. Clearly, $K_2$ is a clique in $G_2$, we show that $K_1 \cup K_2$ is a clique in $G'$. Let $u, v \in K_1 \cup K_2$ with $u \neq v$. If $u, v \in K_1$ or $u, v \in K_2$ then it follows that $\{u, v\} \in E(G')$, so assume without loss of generality that $u = v_i^1$ and $v = v_j^2$ for $i, j \in [n]$. If $i = j$, then $\{v_i^1, v_i^2\} \in E(G')$ by definition. If $i \neq j$, then $v_j^1 \in K_1$. As such, $\{v_i^1, v_j^1\} \in E(G_1)$ since $K_1$ is a clique, and thereby we again have that $\{v_i^1, v_j^2\} \in E(G')$, as desired. Observe that this clique is unique: No size-$\ell'$ clique in $G'$ can contain vertices from $Z$. Furthermore, $G_1$ and $G_2$ have no cliques of size larger than $\ell$. As such, any size-$\ell'$ clique $K$ in $G'$ has the property that $K \cap V(G_1) = \ell$ and $K \cap V(G_2) = \ell$. If $\mathcal{F}$ has exactly one satisfying assignment, then size-$\ell$ cliques are unique in $G_1$ and $G_2$.

Suppose $\mathcal{F}$ is unsatisfiable. Thereby, $G_1$ and $G_2$ do not have a clique of size $\ell$. It is easy to see that the subgraph of $G'$ induced by the vertices of $G_1$ and $G_2$ thereby has no clique of size $\ell' - 1$. Thereby, $Z$ is a unique size-$(\ell' - 1)$ clique in $G'$, concluding the proof of the claim. □

To conclude the proof, let $\mathcal{C}$ be the graph class consisting of all graphs that are obtained via the reduction given in the claim above, when starting from a formula $\mathcal{F}$ that has zero or one satisfying assignments. As such, solving Vertex Cover on $\mathcal{C}$ in polynomial time corresponds to solving Unique-3-SAT in polynomial time, implying $\mathsf{NP} = \mathsf{RP}$. Since any graph in $\mathcal{C}$ has exactly one minimum vertex cover, we obtain that indeed $\beta_{\mathcal{C}} = 1$, as any vertex that is not part of the minimum vertex cover forms a (minimal) blocking set. ■

Observe that graphs in the graph class $\mathcal{C}$ constructed in the proof of Theorem 4.2 are always connected, since they are the complement of a disconnected graph. As such, $\mathcal{C}$ is closed under removing connected components. However, $\mathcal{C}$ is not robust because it is not closed under disjoint union. We can however define $\mathcal{C}'$ to contain all graphs for which all connected components lie in $\mathcal{C}$. Observe that $\mathcal{C}'$ is robust, but that $\beta_{\mathcal{C}'} = 1$ and Vertex Cover is not solvable in polynomial time on $\mathcal{C}' \supseteq \mathcal{C}$ unless $\mathsf{RP} = \mathsf{NP}$.

## 4.4. Reducing the Number of Components Outside the Modulator

As mentioned in the previous subsections, bounded blocking set size is necessary to obtain polynomial kernels for Vertex Cover. Many papers that give polynomial kernels for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator showed that their graph class $\mathcal{C}$ has bounded blocking set size [BS19, FS16, JB13, MRS18]. Some of them used the blocking set size of class $\mathcal{C}$ to bound the number of connected components. More precisely, given an instance $(G, k, X)$ of Vertex Cover with $G - X \in \mathcal{C}$ they showed that one can reduce the number of connected components of $G - X$ to $\mathcal{O}(|X|^{\beta_{\mathcal{C}}+1})$. We will show that one can reduce the number of connected components of $G - X$ to $|X|^{\beta_{\mathcal{C}}}$, as a first step towards proving Theorem 4.3. Here we assume that the class $\mathcal{C}$ is robust in order to guarantee that deletion of connected components of $G - X$ again results in a graph of $\mathcal{C}$. In Section 4.4.1 we discuss suitable conditions so that this component reduction can be done efficiently.

Let $(G, k, X)$ be an instance of Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator. We define the set $\mathcal{X} = \{Z \subseteq X \mid Z \text{ is independent in } G \text{ and } 1 \leq |Z| \leq \beta_{\mathcal{C}}\}$ as the collection of *chunks* of $X$. The intuition of defining the set $\mathcal{X}$ of chunks is to find sets in the modulator $X$ for which at least one vertex must be contained in any optimum vertex cover of $G$. The concept of chunks was first introduced by Jansen and Bodlaender [JB13].

To reduce the number of connected components of $G - X$, we use a result due to Hopcroft and Karp [HK73] which computes a certain crown-like structure in a bipartite graph (see Theorem 2.2 in the Preliminaries).

We construct a bipartite graph $G_B$ to which we will apply Theorem 2.2 to find a set of connected components in $G - X$ that can be safely removed from $G$. We denote the set of connected components in $G - X$ by $\mathcal{F}$. The two parts of the bipartite graph $G_B$ are the set $\mathcal{X}$ of chunks and the set $\mathcal{F}$ of connected components in $G - X$. More precisely, for every chunk $Z \in \mathcal{X}$ and for every connected component $H \in \mathcal{F}$ we add a vertex to the bipartite graph. To simplify notation we denote the vertex of $G_B$ that corresponds to a connected component $H \in \mathcal{F}$ or a chunk $Z \in \mathcal{X}$ by $H$ or $Z$, respectively. We add an edge between a vertex $H \in \mathcal{F}$ and a vertex $Z \in \mathcal{X}$ when $N_G(Z) \cap V(H)$ is a blocking set in $H$, i.e., when $\text{vc}(H - N_G(Z)) + |N_G(Z) \cap V(H)| > \text{vc}(H)$.

It follows from Theorem 2.2 that there exists either a maximum matching $M$ of $G_B$ that saturates $\mathcal{X}$ or a set $\mathcal{X}' \subseteq \mathcal{X}$ such that $|N_{G_B}(\mathcal{X}')| < |\mathcal{X}'|$ and such that there exists a maximum matching $M$ of $G_B - N_{G_B}[\mathcal{X}']$ that saturates $\hat{\mathcal{X}} = \mathcal{X} \setminus \mathcal{X}'$. If there exists a maximum matching $M$ of $G_B$ that saturates $\mathcal{X}$ then let $\mathcal{X}' = \emptyset$ and let $\hat{\mathcal{X}} = \mathcal{X}$. Let $\mathcal{F}_D = \mathcal{F} \setminus (N_{G_B}(\mathcal{X}') \cup V(M))$ be the set of connected components in $\mathcal{F}$ that are neither in the neighborhood of $\mathcal{X}'$ nor endpoint of a matching edge of $M$.

**Reduction Rule 4.1.** Delete all connected components in $\mathcal{F}_D$ from $G$ and decrease the size of $k$ by the size $\text{vc}(\mathcal{F}_D)$ of an optimum vertex cover in $\mathcal{F}_D$.

Observe that Reduction Rule 4.1 also deletes all connected components $H \in \mathcal{F}$ that have the property that for all sets $Z \in \mathcal{X}$ it holds that $N(Z) \cap V(H)$ is not a blocking set of $H$, because these connected components correspond to isolated vertices in the bipartite graph $G_B$. Furthermore, since the minimal blocking set size of each graph in $\mathcal{C}$ is bounded by $\beta_{\mathcal{C}}$, there exists no independent set $Z \subseteq X$ such that $N(Z) \cap V(H)$ is a blocking set in $H$, when $H$ corresponds to an isolated vertex in $G_B$. Before we show the correctness of Reduction Rule 4.1 we show the following lemma which guarantees the existence of certain optimum vertex covers of $G$.

**Lemma 4.5.** *There exists an optimum vertex cover $S$ of $G$ with $S \cap Z \neq \emptyset$ for all chunks $Z \in \hat{\mathcal{X}}$.*

**Proof.** Let $S$ be an optimum vertex cover of $G$. If $S \cap Z \neq \emptyset$ for all $Z \in \hat{\mathcal{X}}$ then we are done, since $S$ fulfills the requirements of the lemma. Thus, assume that there exists at least one chunk $Z \in \hat{\mathcal{X}}$ such that $S \cap Z = \emptyset$. Let $\widetilde{\mathcal{X}} = \{Z \in \hat{\mathcal{X}} \mid S \cap Z = \emptyset\}$ be the set of chunks in $\hat{\mathcal{X}}$ that have an empty intersection with the optimum vertex cover $S$, and let $\widetilde{\mathcal{F}} = \{H \in \mathcal{F} \mid \exists Z \in \widetilde{\mathcal{X}} \colon \{Z, H\} \in M\}$ be the set of connected components in $\mathcal{F}$ that are matched to a vertex in $\widetilde{\mathcal{X}}$ via the matching $M$ that saturates $\hat{\mathcal{X}}$; thus, $|\widetilde{\mathcal{X}}| = |\widetilde{\mathcal{F}}|$.

Claim 4.6. *It holds for all connected components $H \in \widetilde{\mathcal{F}}$ that $|V(H) \cap S| > \mathrm{VC}(H)$.*

Proof. For every connected component $H \in \widetilde{\mathcal{F}}$ let $Z_H \in \widetilde{\mathcal{X}}$ be the chunk that is matched to $H$ via $M$, i.e., $\{H, Z_H\} \in M$. Thus, the set $Y_H = N_G(Z_H) \cap V(H)$ is a blocking set in $H$. Since $Z_H \cap S = \emptyset$, it holds that the vertex cover $S \cap V(H)$ of $H$ must contain the blocking set $Y_H$; hence $|V(H) \cap S| > \mathrm{VC}(H)$. □

Now, we construct an optimum vertex cover $S'$ of $G$ that fulfills the properties of the lemma. First, we add from each chunk $Z \in \widetilde{\mathcal{X}}$ one arbitrary vertex to the set $S$. We denote the resulting set by $\hat{S}$. It holds that $|\hat{S}| \leq |S| + |\widetilde{\mathcal{X}}|$.

Claim 4.7. *For every connected component $H \in \widetilde{\mathcal{F}}$ there exists an optimum vertex cover $S_H$ of $H$ that contains the set $Y_H = N_G(X \setminus \hat{S}) \cap V(H)$.*

Proof. Assume for contradiction that the claim does not hold. This implies that the set $Y_H$ is a blocking set of $H$. Let $Y'_H \subseteq Y_H$ be a minimal blocking set of $H$. Since $H$ is a connected component of a graph of graph class $\mathcal{C}$ it holds that $|Y'_H| \leq \beta_{\mathcal{C}}$. For every vertex $y \in Y'_H$ choose an arbitrary vertex $x \in N_G(y) \cap (X \setminus \hat{S})$ and denote the resulting set by $Z_H$. The set $Z_H$ has size at most $\beta_{\mathcal{C}}$ and is an independent set because it is a subset of the independent set $X \setminus \hat{S}$. Therefore, $Z_H$ is a chunk in $\mathcal{X}$. By construction, the neighborhood of $Z_H$ in $H$ is a superset of $Y'_H$ and thus a blocking set in $H$. It follows that $Z_H$ is a chunk in $\hat{\mathcal{X}}$ because $\{Z_H, H\} \in E(G_B)$, and because $H$ is a vertex in $G_B - N_{G_B}[\mathcal{X}']$. But, by construction, every chunk $Z$ in $\hat{\mathcal{X}}$ has a non-empty intersection with $\hat{S}$. This is a contradiction to the assumption that $Y_H$ is a blocking set of $H$ and proves the claim. □

In a next step, we replace for every connected component $H \in \widetilde{\mathcal{F}}$ the set $S \cap V(H) \subseteq \hat{S}$ by an optimum vertex cover $S_H$ in $H$ that contains the set $N_G(X \setminus \hat{S}) \cap V(H)$. The existence of such an optimum vertex cover $S_H$ follows from Claim 4.7. We denote the resulting set by $S'$. It follows from Claim 4.6 that the set $S'$ has size at most $|S|$ because we replace for each connected component $H \in \widetilde{\mathcal{F}}$ the non-optimum vertex cover $S \cap V(H)$ by the optimum vertex cover $S_H$, more precisely, $|S'| \leq |\hat{S}| - |\widetilde{\mathcal{F}}| \leq |S| + |\widetilde{\mathcal{X}}| - |\widetilde{\mathcal{F}}| = |S|$.

It remains to show that $S'$ is a vertex cover of $G$. We only add vertices of the modulator $X$ to the vertex cover and we change the vertex cover of the connected components in $\widetilde{\mathcal{F}}$. Thus, any edge that is possibly not covered by $S'$ must either be contained in one of the connected components of $\widetilde{\mathcal{F}}$ or between such a connected component and the modulator $X$. Both is not possible, because we add for each connected component $H \in \widetilde{\mathcal{F}}$ a vertex cover $S_H$ to the set $\hat{S}$ which contains the neighborhood of all vertices of $X$ that are not in the vertex cover $\hat{S}$. This concludes the proof. ∎

Now, we show the correctness of Reduction Rule 4.1 using Lemma 4.5. Let $(\widetilde{G}, \widetilde{k}, X)$ be the reduced instance, i.e., $\widetilde{G} = G - \mathcal{F}_D$ and $\widetilde{k} = k - \text{VC}(\mathcal{F}_D)$. Obviously, if $(G, k, X)$ is a yes-instance then $(\widetilde{G}, \widetilde{k}, X)$ is a yes-instance. For the other direction, assume that $(\widetilde{G}, \widetilde{k}, X)$ is a yes-instance. Observe that $M$ is also a matching in $\widetilde{G}_B$ that saturates $\hat{\mathcal{X}}$ because we delete no connected component that is an endpoint of a matching edge. Furthermore, it holds that either $\hat{\mathcal{X}} = \mathcal{X}$ or $|N_{\widetilde{G}_B}(\mathcal{X}')| < |\mathcal{X}'|$ because we delete no connected component that corresponds to a vertex in $N_{\widetilde{G}_B}(\mathcal{X}')$. Thus, it follows from Lemma 4.5 that there exists an optimum vertex cover $\widetilde{S}$ of $\widetilde{G}$ with $\widetilde{S} \cap Z \neq \emptyset$ for all sets $Z \in \hat{\mathcal{X}}$. Note that every connected component $H \in \mathcal{F}_D$ is only adjacent to vertices in $\hat{\mathcal{X}}$ in $G_B$. Since every set $Z \in \hat{\mathcal{X}}$ has a non-empty intersection with the set $\widetilde{S}$, it holds that there exists an optimum vertex cover $S_H$ of $H$ which contains the set $N_G(X \setminus \widetilde{S}) \cap V(H)$ (similar to Claim 4.7). Let $S$ be the set that results from adding for each connected component $H \in \mathcal{F}_D$ the optimum vertex cover $S_H$ to the set $\widetilde{S}$. By construction, it holds that $S$ is a vertex cover of $G$ of size $|\hat{S}| + \text{VC}(\mathcal{F}_D) \leq \hat{k} + \text{VC}(\mathcal{F}_D) = k$. This proves that $(G, k, X)$ is a yes-instance. Overall, we showed that Reduction Rule 4.1 is safe.

**Theorem 4.8.** *Let $(G, k, X)$ be an instance of* Vertex Cover *parameterized by the size of a $\mathcal{C}$-modulator that is reduced under Reduction Rule 4.1. The graph $G - X$ has at most $|X|^{\beta_{\mathcal{C}}}$ connected components.*

**Proof.** Since instance $(G, k, X)$ is reduced under Reduction Rule 4.1 every connected component of $G - X$ is either in $N_{G_B}(\mathcal{X}')$ or an endpoint of a matching edge in $M$. Recall that $M$ is a matching in $G - N_{G_B}[\mathcal{X}']$ that saturates all vertices in $\hat{\mathcal{X}}$. Thus, the number of connected components in $G - X$ is bounded by $|N_{G_B}(\mathcal{X}')| + |\hat{\mathcal{X}}| \leq |\mathcal{X}'| + |\hat{\mathcal{X}}| = |\mathcal{X}|$. The set $\mathcal{X}$ of chunks contains at most $\sum_{i=1}^{\beta_{\mathcal{C}}} \binom{|X|}{i} \leq |X|^{\beta_{\mathcal{C}}}$ many sets; hence $G - X$ has at most $|X|^{\beta_{\mathcal{C}}}$ many connected components. ∎

### 4.4.1. Applying Reduction Rule 4.1 in Polynomial Time

To use Theorem 4.8, we have to prove that we can efficiently reduce the number of connected components in $G - X$ when $X$ is a $\mathcal{C}$-modulator, i.e., that, under certain assumptions, Reduction Rule 4.1 can be applied in polynomial time. We start by providing two sufficient conditions in the next lemma.

**Lemma 4.9.** *Let $\mathcal{C}$ be a graph class where $\beta_{\mathcal{C}}$ is bounded. If* Vertex Cover *is solvable in polynomial time on graphs of class $\mathcal{C}$ and if we can verify in polynomial time whether a given set $Y$ is a blocking set in a graph of class $\mathcal{C}$ then we can apply Reduction Rule 4.1 in polynomial time.*

**Proof.** To apply Reduction Rule 4.1 to an instance $(G, k, X)$ of Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator, we have to construct the bipartite graph $G_B$. Therefore, we have to figure out for each chunk $Z \in \mathcal{X}$ and for every connected component $H$ of $G - X$ whether $N_G(Z) \cap V(H)$ is a blocking set in $H$. We can do this in polynomial time when $\beta_{\mathcal{C}}$ is constant and when we can verify in polynomial time whether $N_G(Z) \cap V(H)$ is a blocking set in $H$. Thus, under the assumptions of the lemma we can construct $G_B$ in polynomial time. Using Theorem 2.2 (Hopcroft Karp [HK73]) we can compute $\mathcal{X}'$, $\hat{X}$ and $\mathcal{F}_D$ in polynomial time. Hence, Reduction Rule 4.1 is applicable in polynomial time. ■

We continue by providing two cases that satisfy the preconditions for the lemma above, such that Reduction Rule 4.1 can be applied in polynomial time on these graph classes.

First of all, we consider the case that graph class $\mathcal{C}$ is hereditary. In this case, being solvable in polynomial time on the class $\mathcal{C}$ is sufficient to also be able to verify whether a given subset of the vertices is a blocking set, thus allowing us to apply Reduction Rule 4.1 in polynomial time. As mentioned in Subsection 4.4 we also need that $\beta_{\mathcal{C}}$ is bounded. Overall, we assume that $\mathcal{C}$ is a hereditary graph class on which Vertex Cover is polynomial-time solvable and where $\beta_{\mathcal{C}}$ is bounded.

**Lemma 4.10.** *Let $\mathcal{C}$ be any hereditary graph class on which* Vertex Cover *can be solved in polynomial time and where $\beta_{\mathcal{C}}$ is bounded. Then Reduction Rule 4.1 can be applied in polynomial time.*

**Proof.** We only have to prove that graph class $\mathcal{C}$ fulfills the requirements of Lemma 4.9. Since we assumed that $\beta_{\mathcal{C}}$ is bounded and that Vertex Cover is polynomial-time solvable on graphs of graph class $\mathcal{C}$, it remains to show that we can verify whether a given vertex set $Y$ is a blocking set of a graph $G$ of graph class $\mathcal{C}$. For this purpose, we compute the size of an optimum vertex cover of $G$ and an optimum vertex cover of $G - Y$. Both is possible in polynomial time because $G$ and $G - Y$ are graphs of graph class $\mathcal{C}$ due to the fact that $\mathcal{C}$ is hereditary. Now, if $\text{vc}(G) = \text{vc}(G - Y) + |Y|$ then $Y$ is not a blocking set of $G$, and if $\text{vc}(G) < \text{vc}(G - Y) + |Y|$ then $Y$ is a blocking set of $G$. Thus, $\mathcal{C}$ fulfills the requirements of Lemma 4.9 which proves that we can apply Reduction Rule 4.1 in polynomial time. ■

Theorem 4.3 (restated below) now follows directly from Theorem 4.8 and Lemmas 4.9 and 4.10.

**Theorem 4.3.** *Let $\mathcal{C}$ be any hereditary graph class with minimal blocking set size $d$ on which* Vertex Cover *can be solved in polynomial time. There is an efficient algorithm that given $(G, k, X)$ such that $G - X \in \mathcal{C}$ returns an equivalent instance $(G', k', X)$ such that $G' - X \in \mathcal{C}$ has at most $\mathcal{O}(|X|^d)$ connected components.*

We can actually further generalize Theorem 4.3 to some non-hereditary graph classes. However, we have more problems to show that Lemma 4.9 holds for non-hereditary graph classes, because after deleting vertices from a graph $G$ that is contained in a non-hereditary graph class $\mathcal{C}$ we do not know whether the resulting graph still belongs to the graph class $\mathcal{C}$. As such we need the additional assumption that Vertex Cover is also polynomial-time solvable on the graph class $\mathcal{C}+1$.

This additional assumption is not unreasonable, when our goal is to obtain a kernelization algorithm for Vertex Cover. In fact, in order to obtain any kernel for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator it is necessary to assume that the problem is fixed-parameter tractable. From this, it immediately follows that we can solve Vertex Cover in polynomial time on $\mathcal{C}+1$.

**Lemma 4.11.** *Let $\mathcal{C}$ be a (possibly non-hereditary) graph class. If we can solve* Vertex Cover *in polynomial time on graphs of graph class $\mathcal{C}$ and $\mathcal{C}+1$ then we can verify in polynomial time whether a set $Y \subseteq V(G)$ is a blocking set of $G$.*

**Proof.** Let $G$ be a graph of graph class $\mathcal{C}$ and let $Y \subseteq V(G)$ be any vertex set. We can assume, without loss of generality, that $G$ is connected, because $Y$ is a blocking set of $G$ if and only if for at least one connected component $H$ of $G$ the set $Y \cap V(H)$ is a blocking set of $H$ (see Proposition 3.5 item (iii)). Therefore, it is enough to show that for each connected component $H$ of $G$ we can verify in polynomial time whether $Y \cap Y(H)$ is a blocking set.

We construct the graph $\hat{G}$ by adding one vertex $v_Y$ to the connected graph $G$ and by connecting the vertex $v_Y$ to every vertex in $Y$. Observe that $\hat{G}$ is a connected graph that is contained in graph class $\mathcal{C}+1$, because $\hat{G}-v_H$ is a connected graph of class $\mathcal{C}$. If $Y$ is not a blocking set of $G$ then $\textsc{vc}(\hat{G}) = \textsc{vc}(G)$ because there exists an optimum vertex cover of $G$ that contains $Y$, which is also a vertex cover of $\hat{G}$ as it contains all neighbors of $v_Y$, and because every vertex cover of $\hat{G}$ restricted to $V(G)$ is a vertex cover of $G$. Now, assume that $Y$ is a blocking set of $G$. This implies that $\textsc{vc}(G) < \textsc{vc}(G-Y)+|Y|$. We will show that $\textsc{vc}(\hat{G}) > \textsc{vc}(G)$. Let $\hat{S}$ be an optimum vertex cover of $\hat{G}$. If $v_Y \in \hat{S}$ then $\hat{S} \setminus \{v_Y\}$ is a vertex cover of $G$; thus $\textsc{vc}(G) \leq |\hat{S}|-1$. If $v_Y \notin \hat{S}$ then we know that $Y \subseteq \hat{S}$. This implies that $|\hat{S}| \geq |Y| + \textsc{vc}(G-Y) > \textsc{vc}(G)$ because $\hat{S}$ is a vertex cover of $G$ that contains $Y$; thus $\textsc{vc}(G) < |\hat{S}|$. Overall, we showed that $\textsc{vc}(\hat{G}) > \textsc{vc}(G)$ when $Y$ is a blocking set of $G$.

Thus, to verify whether $Y$ is a blocking set of $G$ we have to compare the size of an optimum vertex cover of $G$ and $\hat{G}$. Since $G$ is a graph of graph class $\mathcal{C}$ and $\hat{G}$ is a graph of graph class $\mathcal{C}+1$ we can compute these optimum vertex covers in polynomial time. This concludes the proof. ∎

**Theorem 4.12.** *Let $\mathcal{C}$ be any robust graph class with minimal blocking set size $d$ on which* Vertex Cover *can be solved in polynomial time. Furthermore, assume that* Vertex Cover *can be solved in polynomial time on graphs of graph class $\mathcal{C}+1$. There is an efficient algorithm that given $(G,k,X)$ such that $G-X \in \mathcal{C}$ returns an equivalent instance $(G',k',X)$ such that $G'-X \in \mathcal{C}$ has at most $\mathcal{O}(|X|^d)$ connected components.*

**Proof.** This result follows directly from Theorem 4.8, Lemma 4.9 and Lemma 4.11. ∎

## 4.5. Summary

First of all, we showed that for all graph classes $\mathcal{C}$ that are closed under disjoint union the existence of a polynomial kernel for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator implies that every graph of class $\mathcal{C}$ has bounded minimal blocking set size. More specifically, we proved that if $\mathcal{C}$ contains a graph which has a minimal blocking set of size $d$ then Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator $X$ does not have a kernel of size $\mathcal{O}(|X|^{d-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. We continued by showing that the converse direction does not hold, i.e., that there exists a graph class $\mathcal{C}$ where every minimal blocking set has size one but it is unlikely that Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator has a polynomial kernel.

Finally, we showed that bounded blocking set size, even if it is not enough to obtain a polynomial kernel, allows us to reduce the number of connected components of $G - X$, where $X$ is a $\mathcal{C}$-modulator of $G$. More precisely, given an instance $(G, k, X)$ of Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator, where $\mathcal{C}$ is a robust graph class with $\beta_{\mathcal{C}} = d$ such that Vertex Cover is solvable in polynomial time on $\mathcal{C}$ and $\mathcal{C} + 1$, then we can reduce, in polynomial time, the number of connected components in $G - X$ to $\mathcal{O}(|X|^d)$.

# CHAPTER 5
# MINIMAL BLOCKING SET SIZES

## 5.1. Introduction

As seen in the previous chapter, minimal blocking sets play an important role for Vertex Cover kernelization, more precisely, we showed that bounded minimal blocking set size of a graph class $\mathcal{C}$ is crucial to obtain a kernel for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator. In this chapter we consider different types of graph classes which are related to graph classes $\mathcal{C}$ where Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator has a (randomized) polynomial kernel. This includes, for example, the class of independent sets, forests, bipartite graphs, or graphs where the size of an optimum LP solution equals the size of a minimum vertex cover. Recall that we showed in Lemma 3.7 that Vertex Cover parameterized by $\ell = \text{VC} - \text{LP}$ can also be considered as Vertex Cover parameterized by the size of a $\mathcal{C}_{\text{LP}}$-modulator.

### 5.1.1. Minimal Blocking Set Size Relative to Linear Elimination Distance

In Section 5.2 we give tight bounds on the minimal blocking set size for some graph classes that have a constant size modulator to a graph class $\mathcal{C}$ where Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator has a polynomial kernel. It turns out that such graph classes admit a "linear" elimination forest to a graph class $\mathcal{C}$, i.e., an elimination forest that is a path. One can say that such graph classes have a linear elimination distance to a graph class $\mathcal{C}$.

The starting point are results of Fomin and Strømme [FS16] which suggest that the border for bounded minimal blocking set size for feedback vertex set-like parameters may be much more interesting than previously expected. Arguably, there is still quite some room between allowing a single cycle per component and allowing an arbitrary number of cycles as long as they share no vertices. Do larger numbers of cycles per component still allow that the graph has bounded minimal blocking set size? Similarly, cycles in the lower bound proof have odd length and it is known that absence of odd cycles is sufficient, i.e., a kernelization for modulators to bipartite graphs is known which implies that bipartite graphs have bounded minimal blocking set size (see Theorem 4.1). Could this be extended to graphs with one or more odd cycles per connected component?

We show that the answers to the above questions are largely positive and provide, essentially, a single proof to cover them. To this end, it is convenient to take the perspective of feedback sets rather than the maximum size of a cycle packing.

We will show that minimal blocking sets in a $d$-quasi-forest have size at most $d+2$, and that this bound is tight (Section 5.2.1). The case for $d = 1$ strengthens the result of Fomin and Strømme [FS16] (as one cycle per component is stricter than feedback vertex set size one). It turns out that our proof directly extends to $d$-quasi-bipartite graphs, proving that their minimal blocking sets similarly have size at most $d+2$ (Section 5.2.2). This bound is also tight.

Motivated by this, we also explore modulators to graphs in which each connected component has vertex cover size at most $d$ plus the size of a minimum fractional vertex cover, which we call $d$-quasi-integral graphs(Section 5.2.3). We show that minimal blocking sets in any $d$-quasi-integral graph have size at most $2d+2$. Also this bound is tight. All lower bounds for the minimal blocking sets are witnessed by the cliques with $d+2$ or $2d+2$ vertices.

### 5.1.2. Minimal Blocking Set Size Relative to Elimination Distances

Recently, Bougeret and Sau [BS19] presented a polynomial kernelization for Vertex Cover parameterized by the size of a modulator $X$ such that $G - X$ has treedepth at most $d$; here $d$ is a fixed constant and the degree of the polynomial in the kernel size depends exponentially on $d$. To get the kernelization, they prove (in different but equivalent terms) that the size of any minimal blocking set in a graph of treedepth $d$ is at most $2^d$, and they give a lower bound of $2^{d-3}$. As our first result here, we determine the exact maximum size of minimal blocking sets in graphs of treedepth $d$ (see below, and see Section 5.3 for all these results).

Instead of graphs with bounded treedepth we consider a more general class of graphs, namely graphs which have constant elimination distance to some graph class $\mathcal{C}$. Recall, that elimination distance generalizes treedepth. For a hereditary graph class $\mathcal{C}$, we determine the exact maximum size of minimal blocking sets in graphs of elimination distance at most $d$ to $\mathcal{C}$, denoted $\beta_\mathcal{C}(d)$, depending on the maximum minimal blocking set size $\beta_\mathcal{C}$ in the class $\mathcal{C}$.

**Theorem 5.1.** *Let $\mathcal{C}$ be a robust hereditary graph class where $\beta_\mathcal{C}$ is bounded. For every integer $d \geq 1$ it holds that*

$$\beta_\mathcal{C}(d) = \begin{cases} 2^{d-1}+1 & \text{, if } \beta_\mathcal{C} = 1, \\ (\beta_\mathcal{C}-1)2^d+1 & \text{, if } \beta_\mathcal{C} \geq 2. \end{cases}$$

The bound for graphs of treedepth at most $d$ is included in the theorem by using that having treedepth at most $d$ is equivalent to having elimination distance at most $d-1$ to the class of independent sets (i.e., graphs of treedepth one), for which all minimal blocking sets have size 1. Concretely, for the class $\mathcal{C}$ of graphs with treedepth $d$ we get $\beta_{\mathcal{C}} = 1$, if $d = 1$ and $\beta_{\mathcal{C}} = 2^{d-2} + 1$ for $d \geq 2$.

The lower bound holds as well for any non-hereditary class $\mathcal{C}$ but we only get a slightly weaker upper bound for such classes and also require a further technical condition for $\mathcal{C}$ called $f$-solid (Definition 5.23). In particular, we get such an upper bound for the class $\mathcal{C}_{\text{LP}}$ mentioned above. Note that if $\mathcal{C}$ has unbounded minimal blocking set size then the same is true for graphs of any bounded elimination distance to $\mathcal{C}$ (irrespective of $\mathcal{C}$ being hereditary or not).

The requirement that class $\mathcal{C}$ is $f$-solid is not unnatural. In particular, every graph class $\mathcal{C}$ where we can bound the minimal blocking set size of graphs that have a $\mathcal{C}$-modulator of size at most $d$ by a function that depends on $d$ (and $\beta_{\mathcal{C}}$) is $f$-solid. Obviously, the bound for $\text{ed}_{\mathcal{C}}(d)$ should be at least as large as the size of minimal blocking set of a graph that has a $\mathcal{C}$-modulator of size at most $d$ to $\mathcal{C}$.

## 5.2. Linear Elimination Distance

To bound the size of a minimal blocking set $Y$ of a graph $G$ that is a $d$-quasi-forest, a $d$-quasi-bipartite graph, or a $d$-quasi-integral graph, we bound the size of a minimum vertex cover of $G$ and $G - Y$. For this, we use an optimum half-integral solution to $\text{LP}(G - Y)$. But, not any optimum half-integral solution is sufficient. We need the existence of a half-integral solution $x$ to $\text{LP}(G - Y)$ for which *every* minimum vertex cover $X$ in $G - Y$ fulfills $V_1 \subseteq X \subseteq V_{\frac{1}{2}} \cup V_1$. This is similar to the result of Nemhauser and Trotter [NT75] and other results about the connection between minimum vertex covers and their fractional LP solutions [AFLS07, BGL02, CC08, HHS82].

**Lemma 5.2.** *Let $G = (V, E)$ be an undirected graph. There exists an optimum half-integral solution $x \in \{0, \frac{1}{2}, 1\}^{|V|}$ to $\text{LP}(G)$ such that for all minimum vertex covers $X$ of $G$ it holds that $V_1^x \subseteq X \subseteq V \setminus V_0^x$.*

**Proof.** Let $x \in \{0, \frac{1}{2}, 1\}^{|V|}$ be an optimum half-integral solution to $\text{LP}(G)$, such that $V_{\frac{1}{2}}$ is maximal; this means, that there exists no optimum half-integral solution $x'$ to $\text{LP}(G)$ such that $V_{\frac{1}{2}}^x \subsetneq V_{\frac{1}{2}}^{x'}$. We will show that every vertex cover $X$ of $G$ with $V_1^x \nsubseteq X$ or $V_0^x \cap X \neq \emptyset$ is not a minimum vertex cover of $G$.

First, we show that for all subsets $V_1' \subseteq V_1^x$ it must hold that the size of the neighborhood of $V_1'$ in $V_0^x$ is strictly larger than the size of $V_1'$, i.e. $|V_0^x \cap N(V_1')| > |V_1'|$. Assume for the sake of contradiction that there exists a set $V_1' \subseteq V_1^x$ such that $|V_0^x \cap N(V_1')| \leq |V_1'|$. We construct a half-integral solution $x'$ to $\text{LP}(G)$ with $V_{\frac{1}{2}}^x \subsetneq V_{\frac{1}{2}}^{x'}$, by assigning a value of $\frac{1}{2}$ to all vertices in $(V_0^x \cap N(V_1')) \cup V_1'$, and a value of $x_v$ to all other vertices. By the choice of $x'$, it holds that $V^x_{\frac{1}{2}} \subsetneq V^{x'}_{\frac{1}{2}}$ and that

$$w(x') = w(x) - |V_1'| + \frac{1}{2}(|V_0^x \cap N(V_1')| + |V_1'|) \leq w(x).$$

It remains to show that $x'$ is indeed a feasible solution to LP$(G)$. Therefore, it suffices to consider edges $\{u, v\}$ of $G$ that have at least one endpoint in $V_1'$, say $v \in V_1'$, because these are the only vertices for which we decrease the value of the half-integral solution $x$ to obtain $x'$. Since $x'_v = \frac{1}{2}$, the constraint $x'_u + x'_v \geq 1$ can only be violated if $x'_u = 0$. But then it must hold that $x_u = 0$ since the only changed values have value $\frac{1}{2}$ in $x'$. This of course means that $u \in V_0^x \cap N(V_1')$ which implies that $x'_u = \frac{1}{2}$; a contradiction. Overall, this shows that $x'$ is an optimum half-integral solution to LP$(G)$ with $V^x_{\frac{1}{2}} \subsetneq V^{x'}_{\frac{1}{2}}$ which contradicts the assumption, and ends the proof that $|V_0^x \cap N(V_1')| > |V_1'|$ for all sets $V_1' \subseteq V_1^x$.

Now, we show that every minimum vertex cover contains the vertex set $V_1^x$. Again, assume for contradiction that there exists a minimum vertex cover $X$ that does not contain the entire set $V_1^x$. Let $V_1' = V_1^x \setminus X \neq \emptyset$. We will show that adding the set $V_1'$ to the vertex cover $X$ and deleting the set $N(V_1') \cap V_0^x$ from the vertex cover $X$ leads to a strictly smaller vertex cover $X'$ of $G$, i.e. $X' = (X \cup V_1') \setminus (N(V_1') \cap V_0^x)$. First we show that $X'$ has smaller cardinality than $X$. Since $X$ is a vertex cover of $G$, we know that $(N(V_1') \cap V_0^x) \subseteq X$ and hence that the cardinality of $X'$ is $|X| + |V_1'| - |N(V_1') \cap V_0^x|$. From the above observation, we know that $|N(V_1') \cap V_0^x| > |V_1'|$ and it follows that $X'$ has strictly smaller cardinality than $X$. To prove that $X'$ is a vertex cover of $G$, it is enough to show that the neighborhood of every vertex $v \in N(V_1') \cap V_0^x$ is contained in $X'$. This holds because $V_0^x$ is an independent set, $N(V_0^x) \subseteq V_1^x$ and $V_1^x \subseteq X'$. Thus, $X'$ is a vertex cover of $G$ with $|X'| < |X|$. This contradicts the assumption that $X$ is a minimum vertex cover of $G$ and proves that every optimum vertex cover of $G$ contains the set $V_1^x$.

It remains to show that no minimum vertex cover $X$ of $G$ contains a vertex of the set $V_0^x$. Let $X$ be a minimum vertex cover of $G$ and assume that $X \cap V_0^x \neq \emptyset$. Let $v \in V_0^x \cap X$. Since $X$ is a minimum vertex cover, there exists a vertex $w \in N(v) \setminus X$, otherwise $X \setminus \{v\}$ would be a vertex cover of $G$ that is strictly smaller than $X$. However, $N(V_0^x) \subseteq V_1^x$ which implies that $w \in N(v) \setminus X \subseteq N(V_0^x) \setminus X \subseteq V_1^x \setminus X$ since $V_0^x$ is an independent set. This contradicts the fact that every optimum vertex cover of $G$ contains the set $V_1^x$, since $w \in V_1^x \setminus X$. This concludes the proof. ■

Next, we will show a property of minimal blocking sets, more precisely, how a LP solution that fulfills the requirements of Lemma 5.2 relates to a minimal blocking set.

**Lemma 5.3.** *Let $G$ be a graph, let $Y$ be a minimal blocking set of $G$, and let $x \in \{0, \frac{1}{2}, 1\}^{|V(G-Y)|}$ be an optimum half-integral solution to* LP$(G - Y)$ *which fulfills the properties of Lemma 5.2. It holds that $N_G(Y) \subseteq V_{\frac{1}{2}} \cup V_1$.*

**Proof.** Since $Y$ is a minimal blocking set of $G$ it holds for all proper subsets $Y' \subsetneq Y$ that $\text{VC}(G - Y') + |Y'| = \text{VC}(G)$. Thus, for all vertices $y \in Y$ there exists a minimum vertex cover $X_y$ in $G$ that contains all vertices of $Y$, except vertex $y$, i.e., $Y \setminus \{y\} \subseteq X_y$. Consider for all vertices $y \in Y$ the vertex sets $X'_y = X_y \setminus Y$. Obviously, the set $X'_y$ is a vertex cover of the graph $G - Y$ for all vertices $y \in Y$. Furthermore, we know that the sets $X'_y$ are minimum vertex covers of the graph $G - Y$ because

$$|X'_y| + |Y \setminus \{y\}| = |X_y| = \text{VC}(G) = \text{VC}(G - Y) + |Y| - 1,$$

which implies that $|X'_y| = \text{VC}(G - Y)$.

The fact that $X'_y$ is a minimum vertex cover of $G - Y$ for all vertices $y \in Y$ implies that $V_1 \subseteq X'_y = X_y \setminus Y \subseteq V_{\frac{1}{2}} \cup V_1$ because $x$ is an optimum half-integral solution to $\text{LP}(G - Y)$ that fulfills the properties of Lemma 5.2. Since the vertex $y$ is not contained in $X_y$ it holds that $N_G(\{y\}) \subseteq X_y$ which implies that $N_G(\{y\}) \setminus Y \subseteq V_{\frac{1}{2}} \cup V_1$ because $X_y \subseteq V_{\frac{1}{2}} \cup V_1 \cup Y \setminus \{y\}$. It follows that $N_G(Y) \subseteq V_{\frac{1}{2}} \cup V_1$. ∎

### 5.2.1. *d*-Quasi-Forest

Using the above lemmata, we can show that every minimal blocking set in a $d$-quasi-forest has size at most $d + 2$. This generalizes the result of Fomin and Strømme [FS16], who showed that a minimal blocking set in a pseudoforest has size at most three. Furthermore, we can show that this bound is tight.

**Theorem 5.4.** *Let $\mathcal{C}_{d\text{-qf}}$ be the class of d-quasi-forests, i.e., every graph in $\mathcal{C}_{d\text{-qf}}$ is a d-quasi-forest. It holds that $\beta_{\mathcal{C}_{d\text{-qf}}} = d + 2$.*

The crucial part of Theorem 5.4 is to prove the upper bound. We will use Lemma 5.2 to prove this bound.

**Lemma 5.5.** *It holds that $\beta_{\mathcal{C}_{d\text{-qf}}} \leq d + 2$*

**Proof.** Let $G$ be a $d$-quasi-forest and let $Y$ be a minimal blocking set of $G$. Since a minimal blocking set is only contained in one connected component of $G$ (Proposition 3.5 item (iii)), we can assume, without loss of generality, that $G$ is connected. We consider an optimum half-integral solution $x \in \{0, \frac{1}{2}, 1\}^{|V(G-Y)|}$ to $\text{LP}(G - Y)$ which fulfills the properties of Lemma 5.2. Since $Y$ is a minimal blocking set of $G$ it follows from Proposition 3.5 item (iv) that $\text{VC}(G - Y) + |Y| = \text{VC}(G) + 1$.

To bound the size of the set $Y$ we try to find a lower bound for the size of a minimum vertex cover of $G - Y$ and an upper bound for the size of a minimum vertex cover of $G$. An obvious lower bound for the size of a minimum vertex cover of $G - Y$ is the value $\text{LP}(G - Y)$ which is equal to $w(x) = |V_1| + \frac{1}{2}|V_{\frac{1}{2}}|$. This leads to the following lower bound for $\text{VC}(G - Y)$:

$$\text{VC}(G - Y) \geq \text{LP}(G - Y) = |V_1| + \frac{1}{2}|V_{\frac{1}{2}}| = |V_1| + \frac{|G - V_0 - V_1|}{2} - \frac{|Y|}{2}. \tag{5.1}$$

Next, we try to find an upper bound for the size of a minimum vertex cover of $G$. Accordingly, we will construct a vertex cover $X$ of $G$ and the size of this vertex cover is an upper bound for the size of a minimum vertex cover of $G$. First of all, we add all vertices from $V_1$ to $X$. Observe that $N_G(V_0) \subseteq V_1$ because $N_{G-Y}(V_0) \subseteq V_1$ and $N_G(Y) \subseteq V_{\frac{1}{2}} \cup V_1$ (Lemma 5.3). Thus, to extend $V_1$ to a vertex cover of $G$ we only have to add a vertex cover of $G - V_0 - V_1$ to $V_1$.

The graph $G - V_0 - V_1$ is a subgraph of the $d$-quasi-forest $G$ and is therefore also a $d$-quasi-forest. Let $Z \subseteq V(G - V_0 - V_1)$ of size at most $d$ such that $G - V_0 - V_1 - Z$ is a forest; such a set exists by the definition of a $d$-quasi-forest. Let $X_F$ be a minimum vertex cover of the forest $G - V_0 - V_1 - Z$. The vertex cover $X_F$ has size at most $\frac{1}{2}|G - V_0 - V_1 - Z|$ because $G - V_0 - V_1 - Z$ is a forest. Now, $Z \cup X_F$ is a vertex cover of $G - V_0 - V_1$ and $X = V_1 \cup Z \cup X_F$ is a vertex cover of $G$. This leads to the following upper bound for $\textsc{vc}(G)$:

$$\begin{aligned}\textsc{vc}(G) \leq |X| = |V_1| + |Z| + |X_F| &\leq |V_1| + |Z| + \frac{|G - V_0 - V_1 - Z|}{2} \\ = |V_1| + \frac{|Z|}{2} + \frac{|G - V_0 - V_1|}{2} &\leq |V_1| + \frac{|G - V_0 - V_1|}{2} + \frac{d}{2} \end{aligned} \tag{5.2}$$

Using the equation $\textsc{vc}(G - Y) + |Y| = \textsc{vc}(G) + 1$ together with the lower bound for $\textsc{vc}(G - Y)$ and the upper bound for $\textsc{vc}(G)$ leads to the desired upper bound for the size of $Y$:

$$\begin{aligned}|V_1| + \frac{|G - V_0 - V_1|}{2} - \frac{|Y|}{2} + |Y| &\overset{(5.1)}{\leq} \textsc{vc}(G - Y) + |Y| = \textsc{vc}(G) + 1 \\ &\overset{(5.2)}{\leq} |V_1| + \frac{|G - V_0 - V_1|}{2} + \frac{d}{2} + 1 \\ \Longrightarrow\ & |Y| \leq d + 2.\end{aligned}$$

This proves that $Y$ has size at most $d + 2$ and concludes the proof. ■

We showed that every minimal blocking set in a $d$-quasi-forest has size at most $d+2$, i.e., $\beta_{\mathcal{C}_{d\text{-qf}}} \leq d + 2$. To prove Theorem 5.4 it remains to show that the bound is tight:

**Proof of Theorem 5.4.** We show the remaining part of Theorem 5.4, namely that the bound is tight.

Consider the connected graph $H = K_{d+2}$. It holds that $H$ is a $d$-quasi-forest, because any $d$ vertices from $H$ are a feedback vertex set. It holds that a minimum vertex cover of a clique $K_h$ contains $h - 1$ vertices, more precisely, every subset of $h - 1$ vertices is a vertex cover of $K_h$. Thus, it holds that $\textsc{vc}(G) = \textsc{vc}(G - Y') + |Y'|$ for all subsets $Y' \subsetneq V(H)$. Therefore, $Y = V(H)$ is the only blocking set of $H$, and hence a minimal blocking set in $H$ of size $d + 2$ which implies that $\beta_{\mathcal{C}_{d\text{-qf}}} \geq d + 2$. This concludes the proof that $\beta_{\mathcal{C}_{d\text{-qf}}} = d + 2$. ■

### 5.2.2. $d$-Quasi-Bipartite Graph

Now, we will show that $d$-quasi-bipartite graphs have the same bound on the size of minimal blocking sets as $d$-quasi-forests. Even the proof is basically the same because a minimum vertex cover of a bipartite graph, as well as a forests, contains at most half of the vertices.

**Theorem 5.6.** *Let $\mathcal{C}_{d\text{-}qb}$ be the class of d-quasi-bipartite graphs, i.e., every graph in $\mathcal{C}_{d\text{-}qb}$ is a d-quasi-bipartite graph. It holds that $\beta_{\mathcal{C}_{d\text{-}qb}} = d + 2$.*

To prove that $\beta_{\mathcal{C}_{d\text{-qf}}} = d + 2$ we use Lemma 5.3 to obtain an upper and lower bound for the size of a minimum vertex cover in $G$ and $G - Y$, respectively, as in the proof of Lemma 5.5.

**Proof.** Let $G$ be a $d$-quasi-bipartite graph, let $Y$ be a minimal blocking set of $G$. As in the proof of Lemma 5.5 we can assume that $G$ is connected and we consider an optimum half-integral solution $x \in \{0, \frac{1}{2}, 1\}^{|V(G-Y)|}$ to $\mathrm{LP}(G - Y)$ which fulfills the properties of Lemma 5.2. Since $x$ fulfills the properties of Lemma 5.2 it follows from Lemma 5.3 that $N_G(Y) \subseteq V_{\frac{1}{2}} \cup V_1$. Let $Z$ be an odd cycle transversal in $G$ of size at most $d$.

Note that the lower bound $\textsc{vc}(G-Y) \geq |V_1| + \frac{1}{2}|G - V_0 - V_1| - \frac{1}{2}|Y|$ (5.1) also holds in this case, because the value of an optimum half-integral solution is always a valid lower bound for the size of an optimum vertex cover.

But also the upper bound $\textsc{vc}(G) \leq |V_1| + \frac{1}{2}|G - V_0 - V_1| + \frac{1}{2}d$ (5.2) holds in this case, because the set $V_1 \cup Z$ together with a vertex cover of the bipartite graph $G - V_0 - V_1 - Z$ is vertex cover of $G$. Note that a minimum vertex cover of a bipartite graph contains at most half of the vertices. Hence, as in the proof of Lemma 5.5, we obtain that $|Y| \leq d + 2$ which implies that $\beta_{\mathcal{C}_{d\text{-qb}}} \leq d + 2$.

It remains to show that the upper bound of $d + 2$ vertices is tight. Recall that the graph $H = K_{d+2}$ has only one blocking set, namely the set $V(H)$. Furthermore, $H$ is a $d$-quasi-bipartite graph which proves that $\beta_{\mathcal{C}_{d\text{-qb}}} \geq d + 2$ and completes the proof. ■

### 5.2.3. $d$-Quasi-Integral Graph

In contrast to $d$-quasi-forests and $d$-quasi-bipartite graphs, where every minimal blocking set is of size at most $d+2$, $d$-quasi-integral graphs have minimal blocking sets of size up to $2d + 2$. Furthermore, the upper bound for the size of a minimum vertex cover of $G$ works differently. We will bound $\textsc{vc}(G)$ by $\textsc{lp}(G) + d$ instead of the size of a vertex cover of $G$.

**Lemma 5.7.** *Let $\mathcal{C}_{d\text{-}qi}$ be the class of d-quasi-integral graphs, i.e., every graph in $\mathcal{C}_{d\text{-}qi}$ is a d-quasi-integral graph. It holds that $\beta_{\mathcal{C}_{d\text{-}qi}} \leq 2d + 2$.*

**Proof.** Let $G$ be an arbitrary $d$-quasi-integral graph, and let $Y$ be a minimal blocking set of $G$. As in the proof of Lemma 5.5 we can assume that $G$ is connected and we consider an optimum half-integral solution $x \in \{0, \frac{1}{2}, 1\}^{|V(G-Y)|}$ to $\mathrm{LP}(G-Y)$ which fulfills the properties of Lemma 5.2.

Observe that the lower bound $\mathrm{VC}(G-Y) \geq |V_1| + \frac{1}{2}|V_{\frac{1}{2}}|$ (5.1) also holds in this case, because the value of an optimum half-integral solution is always a valid lower bound for the size of a vertex cover.

As mentioned above, we will obtain the upper bound for $\mathrm{VC}(G)$ differently. Instead of constructing a vertex cover of $G$ we construct a feasible solution to $\mathrm{LP}(G)$. Since $G$ is a connected $d$-quasi-integral graph, it holds that $\mathrm{VC}(G) \leq \mathrm{LP}(G) + d$. Now, we construct a feasible solution $x'$ to $\mathrm{LP}(G)$ using the optimum half-integral solution $x$ to $\mathrm{LP}(G-Y)$. Let $x'_v = x_v$ for all vertices $v \in V(G) \setminus Y$, and let $x'_v = \frac{1}{2}$ for all vertices $v \in Y$. To prove that $x'$ is indeed a feasible solution to $\mathrm{LP}(G)$ it remains to consider edges that have at least one endpoint in $Y$. Let $v \in Y$ and let $u \in N_G(y)$. Since $N_G(Y) \subseteq V_{\frac{1}{2}} \cup V_1$ (Lemma 5.3) it follows that $u \in Y \cup V_{\frac{1}{2}} \cup V_1$. This implies that $x'_u \geq \frac{1}{2}$ and proves that $x'$ is a feasible solution to $\mathrm{LP}(G)$. This leads to the following upper bound for $\mathrm{VC}(G)$:

$$\mathrm{VC}(G) \leq \mathrm{LP}(G) + d \leq w(x') + d = |V_1| + \frac{1}{2}|V_{\frac{1}{2}}| + \frac{1}{2}|Y| + d \tag{5.3}$$

Again, using the equation $\mathrm{VC}(G-Y) + |Y| = \mathrm{VC}(G) + 1$ together with the lower bound for $\mathrm{VC}(G-Y)$ and the upper bound for $\mathrm{VC}(G)$ leads to the desired bound for the size of $Y$:

$$\begin{aligned} |V_1| + \frac{1}{2}|V_{\frac{1}{2}}| + |Y| &\overset{(5.1)}{\leq} \mathrm{VC}(G-Y) + |Y| = \mathrm{VC}(G) + 1 \\ &\overset{(5.3)}{\leq} |V_1| + \frac{1}{2}|V_{\frac{1}{2}}| + \frac{1}{2}|Y| + d + 1 \\ \implies &|Y| \leq 2d + 2. \end{aligned}$$

This shows that every minimal blocking set of a $d$-quasi-integral graph has size at most $2d+2$, which concludes the proof. ∎

Next, we will show that this bound is also tight, i.e., that $\beta_{\mathcal{C}_{d\text{-qi}}} = 2d + 2$.

**Theorem 5.8.** *It holds that* $\beta_{\mathcal{C}_{d\text{-qi}}} = 2d + 2$.

**Proof.** The upper bound of $2d+2$ follows from Lemma 5.7. For tightness consider the graph $H = K_{2d+2}$. It holds that $\mathrm{VC}(H) \leq \mathrm{LP}(H) + d$, because $\mathrm{VC}(H) = 2d + 1$ and $\mathrm{LP}(H) = \frac{1}{2}|V(H)| = d + 1$. Furthermore, $Y = V(H)$ is the only blocking set of $H$ because $\mathrm{VC}(H - Y') + |Y'| = \mathrm{VC}(H)$ for all proper subsets $Y'$ of $Y$. This shows that $Y = V(H)$ is a minimal blocking set of $H$ of size $d+2$ which implies that $\beta_{\mathcal{C}_{d\text{-qi}}} \geq 2d+2$ and concludes the proof that $\beta_{\mathcal{C}_{d\text{-qi}}} = 2d + 2$. ∎

### 5.2.4. Graphs Where VC = 2LP − MM

We showed that the parameter $\ell = \textsc{vc} - \textsc{mm}$, $\ell = \textsc{vc} - \textsc{lp}$, and $\ell = \textsc{vc} - (2\textsc{lp} - \textsc{mm})$ can also be considered as the size of a $\mathcal{C}_{\text{MM}}$-, $\mathcal{C}_{\text{LP}}$- or $\mathcal{C}_{2\text{LP}-\text{MM}}$-modulator, respectively. Thus, all the graph classes $\mathcal{C}_{\text{MM}}$, $\mathcal{C}_{\text{LP}}$, and $\mathcal{C}_{2\text{LP}-\text{MM}}$ must have bounded minimal blocking set size. We showed already that the class $\mathcal{C}_{\text{LP}}$ has blocking set size at most two. This holds also for the class $\mathcal{C}_{\text{MM}}$, because $\mathcal{C}_{\text{MM}} = \mathcal{C}_{\text{LP}}$: Obviously, $\mathcal{C}_{\text{MM}} \subseteq \mathcal{C}_{\text{LP}}$ because for every graph $G$ it holds that $\textsc{mm}(G) \leq \textsc{lp}(G) \leq \textsc{vc}(G)$. To see that $\mathcal{C}_{\text{LP}} \subseteq \mathcal{C}_{\text{MM}}$ consider a graph $G$ with $\textsc{vc}(G) = \textsc{lp}(G)$. Let $X$ be an optimum vertex cover of $G$. Since $\textsc{lp}(G) = \textsc{vc}(G)$ it holds that $x \in \{0,1\}^{|V(G)|}$ with $x_v = 1$ for all $v \in X$, and $x_v = 0$ for all $v \in V(G) \setminus X$, is an optimum solution to $\text{LP}(G)$. This implies that there exists a matching between $X$ and $V(G) \setminus X$ that saturates $X$. Hence, $\textsc{vc}(G) = |X| \leq \textsc{mm}(G) \leq \textsc{vc}(G)$ which implies that $G \in \mathcal{C}_{\text{MM}}$ and shows that $\mathcal{C}_{\text{MM}} = \mathcal{C}_{\text{LP}}$.

Now, the question is, what is the bound for $\beta_{\mathcal{C}_{2\text{LP}-\text{MM}}}$? Is this bound also two? The answer to the second question is no, because the $K_3$ has minimal blocking set size three and is contained in $\mathcal{C}_{2\text{LP}-\text{MM}}$. We will show that three is the correct tight bound for the minimal blocking set size.

**Lemma 5.9.** *Let $\mathcal{C}_{2\text{LP}-\text{MM}}$ be the class of graphs $G$ with $\textsc{vc}(G) = 2\textsc{lp}(G) - \textsc{mm}(G)$. It holds that $\beta_{\mathcal{C}_{2\text{LP}-\text{MM}}} = 3$.*

**Proof.** Let $G = (V, E)$ be an arbitrary graph that is contained in $\mathcal{C}_{2\text{LP}-\text{MM}}$, and let $Y$ be a minimal blocking set of $G$. It follows from Proposition 3.5 item (i) that $Y$ contains no vertex that is contained in every optimum vertex cover of $G$. Furthermore, if $Y$ contains a vertex that is not contained in any optimum vertex cover of $G$ then $|Y| = 1$ (see Proposition 3.5 item (ii)), and we are done. Hence, in the following we assume that $Y$ contains neither a vertex that is contained in every optimum vertex cover of $G$ nor a vertex that is not contained in any optimum vertex of $G$. This allows us to delete all isolated vertices because isolated vertices are not contained in any optimum solution and therefore they are not contained in $Y$ or influence optimum solutions.

**Reduction Rule 5.1.** Delete all isolated vertices from $G$.

Obviously, deleting isolated vertices does neither change the size of an optimum vertex cover, the size of an optimum LP solution, nor the size of a maximum matching which implies that the resulting graph is still contained in graph class $\mathcal{C}_{2\text{LP}-\text{MM}}$.

We will delete certain vertices from $G$ that are contained in every optimum vertex cover of $G$, and we will show that the set $Y$ is still a minimal blocking set in the resulting graph. The resulting graph has certain properties which allows us to show that $Y$ is contained in at most one factor-critical component of size at least three of the unique Gallai-Edmonds decomposition of the resulting graph.

**Reduction Rule 5.2.** Let $z \in V$ be a vertex that is contained in every optimum vertex cover of $G$ and that fulfills $\textsc{lp}(G - z) + 1 = \textsc{lp}(G)$. Let $x \in \{0, \frac{1}{2}, 1\}^{|V \setminus \{z\}|}$ be an optimum half-integral solution to $\text{LP}(G - z)$ such that $V_{\frac{1}{2}}$ is maximal. Delete $V_0^x \cup V_1^x \cup \{z\}$ from $G$.

Let $G'$ be the graph that results from applying Reduction Rule 5.2 to graph $G$ and vertex $z \in V$. We will show that $G'$ is contained in the graph class $\mathcal{C}_{2\textsc{lp}-\textsc{mm}}$ and that $Y$ is still a minimal blocking set of $G'$. Since $V^x_{\frac{1}{2}}$ is maximal it follows from the proof of Lemma 5.2 that every optimum vertex cover of $G - z$ contains the set $V^x_1$ and no vertex of the set $V^x_0$. This also implies that every optimum vertex cover of $G$ contains the set $V^x_1 \cup \{z\}$ and no vertex of the set $V^x_0$ because $z$ is contained in every optimum vertex cover of $G$ which implies that $X$ is an optimum vertex cover of $G$ if and only if $X \setminus \{z\}$ is an optimum vertex cover of $G - z$. Thus, $Y$ is also a minimal blocking set of $G' = G - V^x_0 - V^x_1 - z$.

Next, we show that $G'$ is a graph of graph class $\mathcal{C}_{2\textsc{lp}-\textsc{mm}}$, i.e., that $\textsc{vc}(G') = 2\textsc{lp}(G') - \textsc{mm}(G')$. It holds that $\textsc{vc}(G') = \textsc{vc}(G) - |V^x_1| - 1$ because $V^x_1 \cup \{z\}$ is contained in every optimum vertex cover of $G$. Furthermore, it holds that $x' = x|_{V(G')}$ is an optimum half-integral solution to LP$(G')$, because every feasible solution to LP$(G')$ can be extended to a feasible solution to LP$(G - z)$ by assigning value 1 to all vertices in $V^x_1$ and value 0 to all vertices in $V^x_0$. Observe that $\textsc{lp}(G') = w(x') = \textsc{lp}(G) - |V^x_1| - 1$ because $\textsc{lp}(G - z) + 1 = \textsc{lp}(G)$ and $w(x') = \textsc{lp}(G - z) - |V^x_1|$. Every maximum matching $M'$ in $G'$ can be extended to a matching in $G$ by adding the matching between $V^x_1 \cup \{z\}$ and $V^x_0$ that saturates $V^x_1$ to $M'$. Such a matching exists because $\hat{x}$, with $\hat{x}_z = 1$ and $\hat{x}_v = x_v$ otherwise, is an optimum half-integral solution to LP$(G)$. This implies that $\textsc{mm}(G') = \textsc{mm}(G) - |V^x_1| - 1$ because it always holds that $\textsc{mm}(G') \geq \textsc{mm}(G) - |V^x_1 \cup \{z\}|$. Overall, we obtain that $\textsc{vc}(G') = 2\textsc{lp}(G') - \textsc{mm}(G')$ which proves that $G' \in \mathcal{C}_{2\textsc{lp}-\textsc{mm}}$.

Now, assume that neither Reduction Rule 5.1 nor Reduction Rule 5.2 is applicable to the graph $G = (V, E)$. Hence, the unique element $x$ in $\{\frac{1}{2}\}^{|V|}$ is an optimum solution to LP$(G)$. For our next Reduction Rule we need the Gallai-Edmonds decomposition of graph $G$. So, let $(D, A, B)$ be the Gaillai-Edmonds decomposition of $G$.

**Reduction Rule 5.3.** Let $z \in D$ be a vertex that is contained in every optimum vertex cover of $G$. Delete vertex $z$ from $G$.

Let $z \in D$ be a vertex that is contained in every optimum vertex cover of $G$. Since $z$ is contained in every optimum vertex cover of $G$ it holds that $Y$ is a minimal blocking set of $G - z$ (Lemma 3.6). To prove correctness of Reduction Rule 5.3, it remains to show that the graph $G - z$ is a graph of graph class $\mathcal{C}_{2\textsc{lp}-\textsc{mm}}$. The vertex $z$ is contained in $D$ which implies that there exists a maximum matching in $G$ that exposes vertex $z$; thus, $\textsc{mm}(G - z) = \textsc{mm}(G)$. Furthermore, $\textsc{vc}(G - z) = \textsc{vc}(G) - 1$ because $z$ is contained in every optimum vertex cover of $G$. Additionally, it holds that $\textsc{lp}(G - z) = \textsc{lp}(G) - \frac{1}{2}$: Obviously, $\textsc{lp}(G - z) \leq \textsc{lp}(G) - \frac{1}{2}$ because $z$ is contained in every optimum vertex cover. Furthermore, if $\textsc{lp}(G - z) = \textsc{lp}(G) - 1$ then vertex $z$ would fulfill the requirements of Reduction Rule 5.2 which would contradict the assumption that $G$ is reduced under Reduction Rule 5.2; hence $\textsc{lp}(G - z) = \textsc{lp}(G) - \frac{1}{2}$. Consequently, it holds that $\textsc{vc}(G - z) = 2\textsc{lp}(G - z) - \textsc{mm}(G - z)$ which proves that $G - z$ is a graph of graph class $\mathcal{C}_{2\textsc{lp}-\textsc{mm}}$.

In the following we assume that neither Reduction Rule 5.1, Reduction Rule 5.2 nor Reduction Rule 5.3 is applicable to the graph $G = (V, E)$. Thus, $x = \{\frac{1}{2}\}^{|V|}$ is an optimum half-integral solution to LP($G$) (Reduction Rule 5.2), and every vertex $z \in D$ is not contained in every optimum vertex cover of $G$ (Reduction Rule 5.3). Let $D_S = \{v \in D \mid v \text{ isolated in } G - A\}$ be the set of vertices in $D$ that are singleton components of $G[D]$.

Claim 5.10. *The graph $G[A \cup D_S]$ has a perfect matching of size $|D_S|$.*

Proof. It holds for all sets $D'_S \subseteq D_S$ that $|N_G(D'_S)| \geq |D'_S|$ (Lemma 3.9 item (i)). This implies that $|D_S| \leq |A|$ and that there exists a matching in $G[A \cup D_S]$ that saturates $D_S$ (Hall's theorem). Let $M'$ be such a matching in $G[A \cup D_S]$ that saturates $D_S$.

Now, we will show that also every vertex of $A$ is saturated by $M'$. Assume for contradiction that there exists a vertex $v \in A$ such that $v \notin V(M')$. Let $S$ be an optimum vertex cover of $G$ that contains vertex $v$. Such a vertex cover exists because $A$ is the neighborhood of the set $D$ and because for every vertex in $D$ there exists an optimum vertex cover of $G$ that does not contain this vertex; otherwise we can apply one of the previous reduction rules. Let $A' = A \cap S$, and let $M$ be a maximum matching in $G$ such that every vertex which is exposed by $M$ is contained in the optimum vertex cover $S$. The existence of such a maximum matching follows from Lemma 3.9 item (ii) because $V_0^x = \emptyset$. Since $\textsc{vc}(G) - (2\textsc{lp}(G) - \textsc{mm}(G)) = 0$ it follows from Lemma 3.9 item (iii) that every matching edge in $M$ has exactly one endpoint in $S$. This implies that $M$ matches the vertices of $A'$ to vertices of $D_S$ because every optimum vertex cover of $G$ contains $\frac{|V(C)|+1}{2}$ vertices of each connected component $C$ of $G[D \setminus D_S]$. Thus, every vertex in $A$, that is matched to a vertex in $D \setminus D_S$ via $M$, is not contained in $S$. Let $D'_S = \{u \in D_S \mid \exists w \in A' \colon \{u, w\} \in M\}$ be the set of vertices in $D_S$ that are matched to a vertex in $A'$ via $M$. By the choice of $M$, it holds that $D'_S \cap S = \emptyset$ and that $|A'| = |D'_S|$. Furthermore, it holds that $N_G(D'_S) = A'$ because $D'_S \cap S = \emptyset$, because $A' = A \cap S$, and because $N_G(D_S) \subseteq A$. This implies that the matching $M'$ that saturates $D_S$ must have all vertices in $A'$ as endpoints and therefore $M'$ must contain vertex $v \in A' \subseteq A$ as an endpoint. Otherwise, $D'_S$ is not saturated by $M'$. This concludes the proof of the claim. ☐

Depending on the intersection of the minimal blocking set $Y$ with the set $D_S$ we show that the set $Y$ has size at most three.

**Case 1:** Assume that $Y \cap D_S = \emptyset$.

Since $D_S$ is an independent set in $G$ and graph $G[A \cup D_S]$ has a perfect matching of size $|D_S|$ it holds that for each optimum vertex cover $X$ in $G$ the set $X \setminus D_S \cup A$ is an optimum vertex cover of $G$. This implies that $A \cap Y = \emptyset$ because $Y \cap D_S = \emptyset$. Furthermore it follows from Lemma 3.6 item (ii) and item (iii), that $Y$ is a minimal blocking set of $G - A$, because there exists an optimum vertex cover of $G$ that contains $A$ and $A \cap Y = \emptyset$. Thus, $Y$ is contained in exactly one connected component of $G - A$ (Proposition 3.5 item (iii)).

First, assume that $Y$ is contained in a connected component $C$ of $G[D]$. Recall that $C$ has size at least three because $Y \cap D_S = \emptyset$. It holds that $\textsc{lp}(C) = \frac{1}{2}|V(C)|$

because $C$ is factor-critical. Furthermore, it holds that $\textsc{vc}(C) = \frac{1}{2}(|V(C)| + 1)$ because $\textsc{vc}(G) = 2\textsc{lp}(G) - \textsc{mm}(G)$. Thus, $\textsc{vc}(C) - \textsc{lp}(C) = \frac{1}{2}$ which implies that $C$ is $\frac{1}{2}$-quasi-integral. It follows from Theorem 5.8 that $|Y| \le 2 \cdot \frac{1}{2} + 2 = 3$ when $Y$ is contained in $C$.

Second, assume that $Y$ is not contained in a connected component of $G[D]$, i.e., $C$ is a connected component of $G[B]$. It holds that $\textsc{vc}(C) = \frac{|V(C)|}{2}$ because $\textsc{vc}(G) = 2\textsc{lp}(G) - \textsc{mm}(G)$ and that $\textsc{lp}(C) = \frac{|V(C)|}{2}$ because $C$ has a perfect matching. This implies that $|Y| \le 2$ because $C$ is a graph of graph class $\mathcal{C}_{\textsc{lp}}$.

**Case 2:** Assume that $Y \cap D_S \neq \emptyset$ and that $Y' = Y \cap (A \cup D_S)$ is a blocking set in $G[A \cup D_S]$.

We will show that every blocking set $\hat{Y}$ of $G[A \cup D_S]$ is also a blocking set of $G$. Let $S$ be an optimum vertex cover of $G$, and let $M$ be a maximum matching in $G$ such that every vertex that is exposed by $M$ is contained in $S$. Again, the existence follows from Lemma 3.9 item (ii) because $V_0^x = \emptyset$. Since $\textsc{vc}(G) = 2\textsc{lp}(G) - \textsc{mm}(G)$ it follows that $S$ contains exactly one endpoint of each matching edge in $M$ (Lemma 3.9 item (iii)). Thus, every optimum vertex cover of $G$ contains at most $|A|$ vertices of the subgraph $G[S \cup D_S]$ which proves that $\hat{Y}$ is also a blocking set of $G$ because $\textsc{vc}(G[A \cup D_S]) = |A|$.

Overall, we showed that $Y = Y'$ is a minimal blocking set in $G[A \cup D_S]$ because every minimal blocking set $\hat{Y} \subseteq Y'$ of $G[A \cup D_S]$ is also a blocking set of $G$. Observe that $\textsc{vc}(G[A \cup D_S]) = |A| = \textsc{lp}(G[A \cup D_S])$ because there exists a perfect matching in $G[A \cup D_S]$. It follows that $|Y| \le 2$ because $G[A \cup D_S]$ is a graph of graph class $\mathcal{C}_{\textsc{lp}}$.

**Case 3:** Assume that $Y \cap D_S \neq \emptyset$ and that $Y' = Y \cap (A \cup D_S)$ is not a blocking set in $G[A \cup D_S]$.

Claim 5.11. *There exists an optimum vertex cover $S'$ of $G[A \cup D_S]$ with $Y' \subseteq S'$ such that for all optimum vertex cover $\hat{S}$ of $G[A \cup D_S]$ with $Y' \subseteq \hat{S}$ it holds that $A \cap \hat{S} \subseteq A \cap S'$.*

Proof. Since $Y'$ is not a blocking set of $G[A \cup D_S]$ there exists an optimum vertex cover $\hat{S}$ of $G[A \cup D_S]$ with $Y' \subseteq \hat{S}$. Let $S_1'$ and $S_2'$ be two optimum vertex covers of $G[A \cup D_S]$ with $Y' \subseteq S_1' \cap S_2'$. We will show that there exists an optimum vertex cover $S'$ of $G[A \cup D_S]$ with $Y' \subseteq S'$ and $(S_1' \cup S_2') \cap A \subseteq S'$.

Let $S' = ((S_1' \cup S_2') \cap A) \cup (S_1' \cap S_2' \cap D_S)$ and let $M'$ be a perfect matching in $G[A \cup D_S]$. First, we prove that $|S'| = \textsc{vc}(G[A \cup D_S]) = |A|$ by showing that every edge $e \in M'$ has exactly one endpoint in $S'$. Let $e = \{u, v\} \in M'$ with $u \in A$ and $v \in D_S$. If the vertex $u$ is not contained in $S'$ then $u$ is neither contained in $S_1'$ nor $S_2'$. This implies that $v$ is contained in $S_1' \cap S_2'$ because $S_1'$ and $S_2'$ are vertex covers of $G[A \cup D_S]$; thus $v \in S'$. Now, if $u$ is contained in $S'$ then $u$ is contained in $S_1' \cup S_2'$. We assume, without loss of generality, that $u \in S_1'$. Since every edge of $M'$ has exactly one endpoint in $S_1'$ it follows that $v$ is not contained in $S_1'$; hence $v \notin S'$. This concludes the proof that $|S'| = |A| = \text{VC}(G[A \cup D_S])$.

Observe that $Y' \subseteq S'$ because $Y' \subseteq S_1' \cap S_2'$ and because $S_1' \cap S_2' \subseteq S'$. Thus, it remains to show that $S'$ is a vertex cover of $G[A \cup D_S]$. Let $e = \{u, v\}$ be an edge in $G[A \cup D_S]$. The edge $e$ has at least one endpoint in $A$ because $D_S$ is an independent set in $G$. Let, without loss of generality, $u \in A$. If $u$ is also contained in $S_1'$ or $S_2'$ then $u$ is also contained in $S'$. Thus, assume that $u$ is contained neither in $S_1'$ nor in $S_2'$. This implies that $v$ is contained in $S_1'$ and $S_2'$ because $S_1'$ and $S_2'$ are optimum vertex covers of $G[A \cup D_S]$. Thus, $v$ is contained in $S'$ because $S'$ contains the intersection of $S_1'$ and $S_2'$. Overall, we showed that $S'$ is an optimum vertex cover of $G[A \cup D_S]$ with $Y' \subseteq S'$ and $(S_1' \cup S_2') \cap A \subseteq S'$. This proves the claim. $\square$

Let $S'$ be an optimum vertex cover of $G[A \cup D_S]$ with $Y' \subseteq S'$ such that for every optimum vertex cover $\hat{S}$ of $G[A \cup D_S]$ with $Y' \subseteq \hat{S}$ it holds that $A \cap \hat{S} \subseteq A \cap S'$ (Claim 5.11). Observe, for each optimum vertex cover $S$ of $G$, that contains the set $Y'$, we can replace the vertex set $S \cap (A \cup D_S)$ by the optimum vertex cover $S'$ because $S \cap (A \cup D_S)$ is an optimum vertex cover of $G[A \cup D_S]$ which implies that $S \cap A \subseteq S' \cap A$. Thus, the set $S \setminus (A \cup D_S) \cup S'$ is an optimum vertex cover of the graph $G$.

Next, we will show that $Y$ has a non-empty intersection with at most one connected component of $G[D \setminus D_S]$. Assume for contradiction that $Y$ has a non-empty intersection with the two connected components $C_1$ and $C_2$ of $G[D \setminus D_S]$. Since $Y$ is a minimal blocking set of $G$ there exists an optimum vertex cover $S_i$ of $G$ such that $Y \setminus V(C_i) \subsetneq Y$ is contained in $S_i$ for $i \in \{1, 2\}$. We can assume, without loss of generality, that $S' \subseteq S_i$ and that $S' \cap A = S_i \cap A$ for $i \in \{1, 2\}$ (Claim 5.11). But this implies that $S = (S_1 \setminus V(C_1)) \cup (S_2 \cap V(C_1))$ is an optimum vertex cover of $G$ that contains $Y$ because every connected component $C$ of $G[D \setminus D_S]$ contains exactly $\frac{|V(C)|-1}{2}$ vertices of an optimum vertex cover of $G$ and is only adjacent to vertices in $A$. Thus, $Y$ intersects at most one connected component of $G[D \setminus D_S]$. Let $C$ be the one connected component of $G[D \setminus D_S]$ that has a non-empty intersection with $Y$.

Finally, we delete all vertices in $D \setminus (D_S \cup V(C))$ from the graph $G$. We denote the resulting graph by $\hat{G}$. Recall that $\text{VC}(\hat{G}) = \text{VC}(G) - \text{VC}(G[D \setminus (D_S \cup V(C))])$ because every optimum solution of $G$ contains an optimum vertex cover of every connected component of $G - A - D_S$ and an optimum vertex cover of $G[A \cup D_S]$.

First, we show that $Y$ is a minimal blocking set of $\hat{G}$ using Claim 5.11. Assume for contradiction that the set $Y$ is not a blocking set of $\hat{G}$. Thus, there exists an optimum vertex cover $\hat{S}$ of $\hat{G}$ that contains $Y$. Since $\hat{S}$ must contain an optimum vertex cover of $G[A \cup D_S]$ that contains $Y'$ we can assume that $\hat{S}$ contains the set $S'$. Now, given any optimum vertex cover $S$ of $G$ that contains the set $S'$, we can replace the optimum vertex cover $V(C) \cap S$ of $C$ by the optimum vertex cover $V(C) \cap \hat{S}$ of $C$. This leads to an optimum vertex cover of $G$ that contains $Y$ which contradicts the assumption that $Y$ is a blocking set of $G$. Hence, $Y$ is a blocking set of $\hat{G}$. Furthermore, $Y$ is a minimal blocking set of $\hat{G}$ because every optimum vertex cover $S$ of $G$ that contains a subset $\widetilde{Y} \subsetneq Y$ restricted to $\hat{G}$ is an optimum vertex cover of $\hat{G}$ that contains $\widetilde{Y}$.

Second, we show that $\text{VC}(\hat{G}) - \text{LP}(\hat{G}) = \frac{1}{2}$. Observe that there exists a maximum matching of size $\frac{|V(\hat{G})|-1}{2}$ in $\hat{G}$. This matching consists of a perfect matching of the graph $\hat{G}[B] = G[B]$, a perfect matching of the graph $G[A \cup D_S]$ and a near perfect matching of the factor-critical component $C$. This implies that $\text{LP}(\hat{G}) = \text{MM}(\hat{G}) + \frac{1}{2}$ and that $\text{VC}(\hat{G}) = \text{MM}(\hat{G}) + 1$; hence $\text{VC}(\hat{G}) - \text{LP}(\hat{G}) = \frac{1}{2}$. Thus, the graph $\hat{G}$ is $\frac{1}{2}$-quasi-integral and it follows from Theorem 5.8 that the minimal blocking set $Y$ of $\hat{G}$ has size at most $2 \cdot \frac{1}{2} + 2 = 3$.

Overall, we proved that the minimal blocking set $Y$ of $G$ has size at most 3 which shows that $\beta_{\mathcal{C}_{2\text{LP}-\text{MM}}} \leq 3$. It remains to show that $\beta_{\mathcal{C}} \geq 3$. Therefore, consider the graph $G$ that consists of three vertices which are pairwise adjacent, i.e., $G$ is a triangle. It holds that $\text{VC}(G) = 2$, that $\text{LP}(G) = \frac{3}{2}$, and that $\text{MM}(G) = 1$. Thus, the graph $G$ belongs to class $\mathcal{C}_{2\text{LP}-\text{MM}}$ because $\text{VC}(G) - (2\text{LP}(G) - \text{MM}(G)) = 2 - (2 \cdot \frac{3}{2} - 1) = 0$. The only blocking set of $G$ is the entire vertex set $V(G)$ of size three because any two vertices are an optimum vertex cover of $G$. This shows that $\beta_{\mathcal{C}_{2\text{LP}-\text{MM}}} \geq 3$ and concludes the proof that $\beta_{\mathcal{C}_{2\text{LP}-\text{MM}}} = 3$. ■

## 5.3. Minimal Blocking Sets in Graphs of Bounded Elimination Distance

In this section we consider the minimal blocking set size of graphs that have elimination distance $d$ to a given robust graph class $\mathcal{C}$ that has bounded minimal blocking set size. Recall, that graphs of treedepth $d$ as well as graphs which have a $\mathcal{C}$-modulator of size $d$ have elimination distance at most $d$. To obtain an upper bound for the value $\beta_{\mathcal{C}}(d)$ we will show that we can bound the size of a minimal blocking set $Y$ of a connected graph $G$ with $\text{ed}_{\mathcal{C}}(G) = d$ and root $r$ of an optimum elimination forest of $G$ using the size of minimal blocking sets of subgraphs of $G - r$. The lower bound construction shows how one can combine two minimal blocking sets of graphs with elimination distance $d - 1$ to graph class $\mathcal{C}$ to a larger minimal blocking set of a graph with elimination distance $d$ to graph class $\mathcal{C}$.

For hereditary graph classes, these bounds turn out to be tight. Theorem 5.1 will follow directly from the lower bound presented in Theorem 5.12, combined with the upper bound in Theorem 5.17. For non-hereditary graph classes the lower bound construction still holds, but the upper bound is worse and we need to assume an additionally property of the graph class $\mathcal{C}$.

### 5.3.1. Lower Bound

In this subsection we give a lower bound for $\beta_{\mathcal{C}}(d)$ for all robust graph classes $\mathcal{C}$. Our result improves the lower bound for the size of largest minimal blocking sets of graphs with treedepth $d$ given by Bougeret and Sau [BS19], because a graph that has treedepth $d$ has elimination distance $d-1$ to the graph class where every graph is an independent set. We will show how we can glue two graphs together to obtain larger minimal blocking sets.

**Theorem 5.12.** *Let $\mathcal{C}$ be a robust graph class where $\beta_{\mathcal{C}}$ is bounded. For every integer $d \geq 1$ it holds:*

$$\beta_{\mathcal{C}}(d) \geq \begin{cases} 2^{d-1}+1 & \text{, if } \beta_{\mathcal{C}} = 1 \\ (\beta_{\mathcal{C}}-1)2^d+1 & \text{, if } \beta_{\mathcal{C}} \geq 2 \end{cases}$$

To show Theorem 5.12 we construct for each graph class $\mathcal{C}$ where $\beta_{\mathcal{C}}$ is bounded and each integer $d \geq 1$ a graph $G$ with $\mathrm{ed}_{\mathcal{C}}(G) = d$ that contains a minimal blocking set of size at least $2^{d-1}+1$ when $\beta_{\mathcal{C}} = 1$, and of size at least $(\beta_{\mathcal{C}}-1)2^d+1$ when $\beta_{\mathcal{C}} \geq 2$. Since $\beta_{\mathcal{C}}(d) = \max\{\beta(G) \mid \mathrm{ed}_{\mathcal{C}}(G) \leq d\}$, this will prove Theorem 5.12. We use the following two constructions to obtain such a graph $G$.

**Lemma 5.13.** *Let $H = (V, E)$ be a graph and let $Y$ be a minimal blocking set of $H$. Let $H' = (V \cup \{v\}, E \cup \{\{v, y\} \mid y \in Y\})$ be the graph that results from $H$ by adding a new vertex $v$ to $H$ and by connecting it to all vertices in $Y$. It holds that $\textsc{vc}(H') = \textsc{vc}(H)+1$, and that $Y' = Y \cup \{v\}$ is a minimal blocking set of $H'$.*

**Proof.** First, we show that $\textsc{vc}(H') = \textsc{vc}(H) + 1$. It is clear that $\textsc{vc}(H') \leq \textsc{vc}(H) + 1$ because every (optimum) vertex cover of $H$ together with the newly added vertex $v$ is a vertex cover of $H'$. Now, assume for contradiction that $\textsc{vc}(H') = \textsc{vc}(H)$. This implies that every optimum vertex cover $X$ of $H'$ does not contain the vertex $v$, and therefore contains the neighborhood $Y$ of $v$ in $H'$. This contradicts the fact that $Y$ is a blocking set in $H$, because $X$ is an optimum vertex cover of $H$ with $Y \subseteq X$. Thus, $\textsc{vc}(H') = \textsc{vc}(H) + 1$.

Second, we will show that $Y \cup \{v\}$ is a minimal blocking set of $H'$. The set $Y \cup \{v\}$ is a blocking set of $H'$. Otherwise, there would exist an optimum vertex cover $X'$ of $H'$ with $Y \cup \{v\} \subseteq X'$. But this would imply that $X = X' \setminus \{v\}$ is an optimum vertex cover of $H$ with $Y \subseteq X$ which contradicts the assumption that $Y$ is a blocking set of $H$.

Now, assume that $Y \cup \{v\}$ is not a minimal blocking set of $H'$. Hence, there exists a set $Y' \subsetneq Y \cup \{v\}$ that is a minimal blocking set in $H'$. Observe that there is an optimum vertex cover of $H'$ containing $v$ (take a vertex cover in $H$ and add $v$) and an optimum vertex cover containing $N(v)$, by taking a vertex cover in $H$ of size $\textsc{vc}(H)+1$ which contains $Y$. Thereby, it holds that neither the set $\{v\}$ nor the set $N(v) = Y$ is a blocking set of $H'$. Hence, it holds that $Y' \setminus \{v\} \neq \emptyset$ and $v \in Y'$. The set $Y' \setminus \{v\} \subsetneq Y$ is not a blocking set of $H$ because $Y$ is a minimal blocking set of $H$. Thus, there exists an optimum vertex cover $X$ of $H$ that contains the set $Y' \setminus \{v\}$. But, $X \cup \{v\}$ is an optimum vertex cover of $H'$ that contains $Y'$ which implies that $Y'$ is not a blocking set of $H'$ and contradicts the choice of $Y'$. This proves that $Y \cup \{v\}$ is a minimal blocking set of $H'$ and concludes the proof. ■

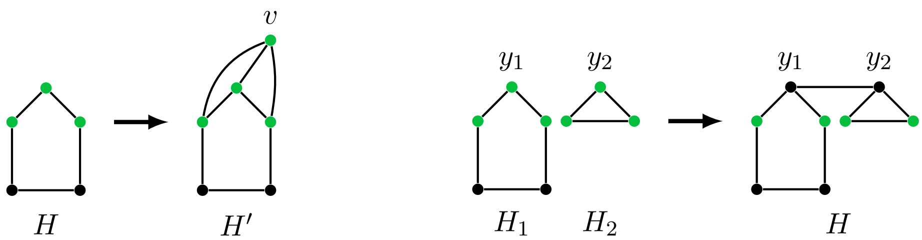
Observation 5.14. It follows from Lemma 5.13 that $\beta_{\mathcal{C}}(d-1) < \beta_{\mathcal{C}}(d)$.

**Lemma 5.15.** *Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two graphs with $\beta(H_1) \geq 2$ and $\beta(H_2) \geq 1$. Let $Y_1$ be a minimal blocking set of $H_1$ of size at least two, let $Y_2$ be a minimal blocking set of $H_2$, and let $y_1 \in Y_1$ and $y_2 \in Y_2$ be two vertices. Let $H = (V_1 \cup V_2, E_1 \cup E_2 \cup \{\{y_1, y_2\}\})$ be the graph that results from the union of the graphs $H_1$ and $H_2$ by additionally connecting the vertices $y_1$ and $y_2$. It holds that $\text{VC}(H) = \text{VC}(H_1) + \text{VC}(H_2)$, and that $Y = (Y_1 \cup Y_2) \setminus \{y_1, y_2\}$ is a minimal blocking set of the graph $H$.*

**Proof.** Since every (optimum) vertex cover of $H$ contains a vertex cover of $H_1$ and $H_2$ it holds that $\text{VC}(H) \geq \text{VC}(H_1)+\text{VC}(H_2)$. The inequality $\text{VC}(H) \leq \text{VC}(H_1)+\text{VC}(H_2)$ follows from the fact that there exists an optimum vertex cover $X_1$ of $H_1$ that contains $y_1$. Such a vertex cover $X_1$ of $H_1$ exists because $Y_1$ is a minimal blocking set of $H_1$ of size at least two that contains the vertex $y_1$. Now, the optimum vertex cover $X_1$ of $H_1$ that contains $y_1$ together with any optimum vertex cover of $H_2$ is a vertex cover of $H$. Thus, $\text{VC}(H) \leq \text{VC}(H_1) + \text{VC}(H_2)$ which concludes the proof that $\text{VC}(H) = \text{VC}(H_1) + \text{VC}(H_2)$.

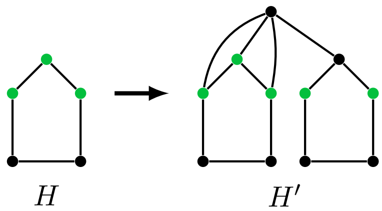
Next, we show that $Y = (Y_1 \cup Y_2) \setminus \{y_1, y_2\}$ is a minimal blocking set of $H$. First, we show that $Y$ is a blocking set of $H$. Assume for contradiction that $Y$ is not a blocking set of $H$. This implies that there exists an optimum vertex cover $X$ of $H$ with $Y \subseteq X$. Since $\{y_1, y_2\}$ is an edge in $H$ it holds that $y_1 \in X$ or $y_2 \in X$. This implies that $Y_1 \subseteq X$ or $Y_2 \subseteq X$. But, this contradicts the assumption that $Y_1$ is a blocking set of $H_1$ or the assumption that $Y_2$ is a blocking set of $H_2$: The sets $X \cap V(H_1)$ and $X \cap V(H_2)$ are optimum vertex covers of $H_1$ or $H_2$, respectively. Furthermore, at least one of these sets contains the vertex set $Y_1$ or $Y_2$ because $Y_1 \subseteq X$ or $Y_2 \subseteq X$. This implies that one of the sets $Y_1$, $Y_2$ is not a blocking set in $H_1$ or $H_2$, respectively, and proves that $Y$ is a blocking set of $H$.

Second, we show that $Y$ is also a minimal blocking set of $H$. Let $Y' \subseteq Y$ be a minimal blocking set of $H$. If $Y' = Y$ then we are done. So, suppose there exists a vertex $y \in Y \setminus Y'$. The vertex $y$ is either contained in $Y_1 \setminus \{y_1\}$ or $Y_2 \setminus \{y_2\}$. Let $i \in \{1, 2\}$ such that $y \in Y_i$, and let $X_i$ be an optimal vertex cover in $H_i$ containing $Y_i \setminus \{y\}$. Let $j \in \{1, 2\} \setminus \{i\}$ and let $X_j$ be an optimal vertex cover in $H_j$ containing $Y_j \setminus \{y_j\}$. Both vertex covers exist, since $Y_i$ and $Y_j$ are minimal blocking sets in $H_i$ or $H_j$, respectively, and since $y \in Y_i$ and $y_j \in Y_j$. Observe that $X_i \cup X_j$ is a vertex cover of $H$, because the set $X_i$ is a vertex cover in $H_i$, the set $X_j$ is a vertex cover in $H_j$ and the edge $\{y_1, y_2\}$ is covered by $y_i$. Furthermore, $X_i \cup X_j$ is a minimum vertex cover of $G$ since $\text{VC}(H) = \text{VC}(H_1) + \text{VC}(H_2)$. However, $Y' \subseteq (Y_1 \cup Y_2) \setminus \{y, y_j\} \subseteq X_i \cup X_j$, contradicting that $Y'$ is a blocking set of $H$. ■




(a) Construction used to prove Lemma 5.13, where we add vertex $v$ to the graph $H$. It holds $\mathrm{ed}_{\mathcal{C}}(H') \leq \mathrm{ed}_{\mathcal{C}}(H) + 1$.

(b) Construction used to prove Lemma 5.15, where we add edge $\{y_1, y_2\}$ to the union of $H_1$ and $H_2$.




(c) Construction used to prove Lemma 5.16, where graph $H'$ consists of two copies of $H$, where we apply construction (a) to one copy and then construction (b) to the two graphs. It holds that $\mathrm{ed}_{\mathcal{C}}(H') \leq \mathrm{ed}_{\mathcal{C}}(H) + 1$.

Figure 5.1.: This figure depicts the construction used to prove Lemmas 5.13, 5.16 and 5.16. The green vertices form minimal blocking sets.

The next lemma shows how we can combine these two constructions to obtain larger minimal blocking sets for a graph of elimination distance $d$ to a graph class $\mathcal{C}$ when given a graph of elimination distance $d-1$ to a graph class $\mathcal{C}$.

**Lemma 5.16.** *Let $H = (V, E)$ be a graph, and let $Y$ be a minimal blocking set of $H$. There exists a graph $H'$ that fulfills the following properties:*

*(i) The graph $H'$ has a minimal blocking set of size $2|Y| - 1$.*

*(ii) For every graph class $\mathcal{C}$ it holds that $\mathrm{ed}_{\mathcal{C}}(H') \leq \mathrm{ed}_{\mathcal{C}}(H) + 1$.*

To prove Lemma 5.16 we combine the constructions of Lemma 5.13 and Lemma 5.15. The combination of these two constructions is depicted in Figure 5.1.

**Proof.** To construct graph $H'$, we first apply Lemma 5.13 to graph $H$ and the minimal blocking set $Y$. Let $H_1$ be the resulting graph, and let $y_1$ be the vertex that we added to $H$ during the construction of Lemma 5.13 to obtain graph $H_1$. It follows from Lemma 5.13 that $Y_1 = Y \cup \{y_1\}$ is a minimal blocking set of $H_1$.

Second, we apply Lemma 5.15 to the two graphs $H_1$ and $H_2 = H$, the minimal blocking set $Y_1 = Y \cup \{y_1\}$ of $H_1$ of size at least two, the minimal blocking set $Y_2 = Y$

of $H_2$, and the vertex $y_1 \in Y_1$ as well as an arbitrary vertex $y_2 \in Y_2$. Let $H'$ be the resulting graph. The set $Y' = (Y_1 \cup Y_2) \setminus \{y_1, y_2\}$ is a minimal blocking set of $H'$ and $Y'$ has size $|Y_1| + |Y_2| - 2 = 2|Y| - 1$ (Lemma 5.15). This proves item (i). Item (ii) holds, because $H' - y_1$ consists of two copies of $H$ that are not connected by any edge. ■

Finally, we are able to prove Theorem 5.12 by induction using the construction of Lemma 5.13 as well as Lemma 5.16. We need the construction of Lemma 5.13 only for the base case of the induction because for a graph with minimal blocking set size one, this construction leads to a graph with minimal blocking set size at least two whereas Lemma 5.16 only leads to a graph of minimal blocking set size one.

**Proof of Theorem 5.12.** We prove Theorem 5.12 by induction over the elimination distance $d$ to graph class $\mathcal{C}$. In the base case of the induction we construct a graph $G$ that has elimination distance at most $d = 1$ to graph class $\mathcal{C}$. Let $H$ be a graph of class $\mathcal{C}$ with $\beta(H) = \beta_{\mathcal{C}}$, and let $Y$ be a minimal blocking set of $H$ of size $\beta_{\mathcal{C}}$. If $\beta_{\mathcal{C}} = 1$ then we apply Lemma 5.13 to graph $H$ and the minimal blocking set $Y$ of $H$ of size $\beta_{\mathcal{C}}$ to obtain graph $G$. Let $y$ be the vertex that we add to graph $H$ during the construction of Lemma 5.13 to obtain graph $G$. Obviously, it holds that $\mathrm{ed}_{\mathcal{C}}(G) \leq 1$. Furthermore, $Y' = Y \cup \{y\}$ is a minimal blocking set of $G$ which implies that $\beta(G) \geq |Y'| = |Y| + 1 = \beta(H) + 1 = \beta_{\mathcal{C}} + 1 = 2 = 2^{1-1} + 1$. Hence, $\beta_{\mathcal{C}}(1) \geq \beta(G) \geq 2^{1-1} + 1$ for $d = 1$ and $\beta_{\mathcal{C}} = 1$ which is the desired bound for Theorem 5.12.

In the case that $\beta_{\mathcal{C}} \geq 2$ we apply Lemma 5.16 to a graph $H \in \mathcal{C}$ with $\beta(H) = \beta_{\mathcal{C}}$, and a minimal blocking set $Y$ of $H$ of size $\beta_{\mathcal{C}}$ to obtain graph $G$. It follows from Lemma 5.16 item (ii) that $\mathrm{ed}_{\mathcal{C}}(G) \leq \mathrm{ed}_{\mathcal{C}}(H) + 1 = 1$ and from item (i) that $\beta_{\mathcal{C}}(d) \geq \beta(G) \geq 2|Y| - 1 = 2\beta_{\mathcal{C}} - 1 = (\beta_{\mathcal{C}} - 1) \cdot 2^1 + 1$. This concludes the proof of Theorem 5.12 for $d = 1$.

For the induction step, we assume that the statement is true for all integers less than $d$, and all graph classes $\mathcal{C}$ where $\beta_{\mathcal{C}}$ is bounded.

By induction hypothesis, there exists a graph $H$ with $\mathrm{ed}_{\mathcal{C}}(H) \leq d - 1$ such that

$$\beta(H) \geq \begin{cases} 2^{(d-1)-1} + 1 & \text{, if } \beta_{\mathcal{C}} = 1 \\ (\beta_{\mathcal{C}} - 1)2^{(d-1)} + 1 & \text{, if } \beta_{\mathcal{C}} \geq 2 \end{cases} \tag{5.4}$$

because $\beta_{\mathcal{C}}(d-1)$ has at least this size. Let $Y$ be a minimal blocking set of $H$ of size $\beta(H)$. Again, we apply Lemma 5.16 to the graph $H$ and the minimal blocking set $Y$. It follows from Lemma 5.16 item (ii) that $\mathrm{ed}_{\mathcal{C}}(G) \leq \mathrm{ed}_{\mathcal{C}}(H) + 1 = d$ and from item (i) that $\beta(G) \geq 2|Y| - 1 = 2\beta(H) - 1$. Using the induction hypothesis (inequality 5.4) we obtain:

$$\beta(G) \geq 2\beta(H) - 1 \overset{(5.4)}{\geq} \begin{cases} 2 \cdot \left(2^{(d-1)-1} + 1\right) - 1 = 2^{d-1} + 1 & \text{, if } \beta_{\mathcal{C}} = 1 \\ 2 \cdot \left((\beta_{\mathcal{C}} - 1)2^{(d-1)} + 1\right) - 1 = (\beta_{\mathcal{C}} - 1)2^d + 1 & \text{, if } \beta_{\mathcal{C}} \geq 2. \end{cases}$$

This concludes the proof of Theorem 5.12 because $\beta_{\mathcal{C}}(d) = \max\{\beta(G) \mid \mathrm{ed}_{\mathcal{C}}(G) \leq d\}$. ■

### 5.3.2. Upper Bound

We will show that our lower bound for $\beta_{\mathcal{C}}(d)$, which we proved in Theorem 5.12, is tight when the graph class $\mathcal{C}$ is hereditary. As mentioned above, this gives us also a tight bound for the size of a largest minimal blocking sets for graphs with treedepth $d$. Thus, our result improves the upper and lower bound for the largest minimal blocking set size of graphs of treedepth (at most) $d$ [BS19]. Afterwards, we will show an upper bound for $\beta_{\mathcal{C}}(d)$ for some non-hereditary graph classes, including the class of graphs where the optimum solution is equal to the optimum LP solution. This bound is weaker than the bound for $\beta_{\mathcal{C}}(d)$ when $\mathcal{C}$ is hereditary.

**Hereditary graph classes.** To prove the tight lower bound for robust hereditary graph classes we bound the size of a minimal blocking set $Y$ in a graph $G$ using a minimal blocking set $Y'$ of a subgraph $G'$ of $G$ that has smaller elimination distance to the given graph class. The choice of $G'$ and $Y'$ depends on, among other things, whether the root $r$ of an optimum elimination forest of $G$ is contained in $Y$ or not.

**Theorem 5.17.** *Let $\mathcal{C}$ be a robust hereditary graph class where $\beta_{\mathcal{C}}$ is bounded, and let $d \geq 0$. Then*

$$\beta_{\mathcal{C}}(d) \leq \begin{cases} 1 & \text{, if } \beta_{\mathcal{C}} = 1 \text{ and } d = 0, \\ 2^{d-1} + 1 & \text{, if } \beta_{\mathcal{C}} = 1 \text{ and } d \geq 1, \\ (\beta_{\mathcal{C}} - 1)2^d + 1 & \text{, if } \beta_{\mathcal{C}} \geq 2. \end{cases}$$

We consider different cases in our proof. But, only one case needs the requirement that the class $\mathcal{C}$ is hereditary. We will see later that we can bound the blocking set size also for some non-hereditary class $\mathcal{C}$ for the remaining case.

**Lemma 5.18.** *Let $G$ be a graph, let $Y$ be a minimal blocking set in $G$, and let $r \in Y$ be an arbitrary vertex that is contained in the blocking set $Y$.*

*(i) If $\mathrm{VC}(G) = \mathrm{VC}(G - r)$ then $Y = \{r\}$.*

*(ii) If $\mathrm{VC}(G) = \mathrm{VC}(G - r) + 1$ then $Y \setminus \{r\}$ is a minimal blocking set of $G - r$.*

**Proof.** If $\mathrm{VC}(G) = \mathrm{VC}(G - r)$ then no optimum vertex cover of $G$ contains $r$. Thus, it follows from Proposition 3.5 item (ii) that $\{r\}$ is a (minimal) blocking set of $G$ which implies that $Y = \{r\}$ and proves item (i).

Now, assume that $\mathrm{VC}(G) = \mathrm{VC}(G - r) + 1$. Consequently, there exists an optimum vertex cover of $G$ that contains $r$ which implies that $Y \neq \{r\}$. Let $Y' = Y \setminus \{r\} \neq \emptyset$. It follows from Lemma 3.6 item (ii) that $Y'$ is a blocking set of $G - r$. Furthermore, it follows from Lemma 3.6 item (iii) that $Y'$ is a minimal blocking set of $G - r$, because every minimal blocking set $\hat{Y} \subseteq Y'$ of $G - r$ can be extended to the blocking set $\hat{Y} \cup \{r\} \subseteq Y$ of $G$. Since $Y$ is a minimal blocking set of $G$, the set $Y'$ must be a minimal blocking set of $G - r$. ∎

**Lemma 5.19.** *Let $G$ be a graph that contains a vertex $r$ that is not contained in any optimum vertex cover of $G$; hence $\text{VC}(G-r) = \text{VC}(G)$. Let $Y$ be a minimal blocking set of $G$ with $r \notin Y$.*[1] *There exists a (possibly empty) set $Z \subseteq N(r)$ such that $Y \cup Z$ is a minimal blocking set of $G - r$.*

**Proof.** Observe that $Y$ may not be a blocking set of $G-r$ because $r$ forces the vertices in $N(r)$ to be in every optimum vertex cover of $G$. However, the set $Y \cup N(r)$ is a blocking set of $G-r$. Otherwise, there exists an optimum vertex cover $X$ of $G-r$ that contains $Y \cup N(r)$. But, the set $X$ is also an optimum vertex cover of $G$ that contains $Y$ which contradicts the assumption that $Y$ is a blocking set of $G$.

Let $Y' \subseteq Y \cup N(r)$ be a minimal blocking set of $G - r$. Note that $Y' \setminus N(r) \neq \emptyset$ because every optimum vertex cover of $G$ contains $N(r)$ which implies that there exists an optimum vertex cover of $G - r$ that contains $N(r)$. We will show that the minimal blocking set $Y'$ must contain the set $Y$, by proving that the non-empty set $Y' \setminus N(r)$ is a blocking set of $G$. If the set $Y' \setminus N(r)$ is not a blocking set of $G$ then there exists an optimum vertex cover $X$ of $G$ that contains the set $Y' \setminus N(r)$. However, the set $X$ contains also the vertex set $N(r)$ because every optimum vertex cover of $G$ does not contain $r$. Thus, $Y' \subseteq X$ and $X$ is also an optimum vertex cover of $G - r$ which contradicts the fact that $Y'$ is a blocking set of $G - r$. Now, the set $Z = Y' \cap N(r)$ fulfills the desired properties. ■

**Lemma 5.20.** *Let $G$ be a graph that contains a vertex $r$ that is contained in every optimum vertex cover of $G$; hence $\text{VC}(G-r)+1 = \text{VC}(G)$. Let $Y$ be a minimal blocking set of $G$. The set $Y$ is also a minimal blocking set of $G - r$.*

**Proof.** Since $r$ is contained in every optimum vertex cover of $G$ it follows that $r$ is not contained in any minimal blocking set of $G$, and therefore $r \notin Y$ (Proposition 3.5 item (i)). Furthermore, it follows from Lemma 3.6 item (ii) that $Y$ is also a blocking set of $G - r$. Item (iii) of Lemma 3.6 implies that $Y$ is also a minimal blocking set of $G - r$ because no minimal blocking set of $G$ contains vertex $r$. ■

So far, we showed that given a graph $G$, a vertex $r$, and a minimal blocking set $Y$ of $G$ we can bound the size of $Y$ by $\beta(G - r) + 1$, except when $r$ is contained in at least one, but not all, optimum vertex covers of $G$, and $r \notin Y$. In this case it is possible that neither $Y$ nor a superset of $Y$ is a blocking set of $G - r$. The following lemma provides some properties of $Y$ that helps us to bound the size of $Y$.

**Lemma 5.21.** *Let $G$ be a graph that contains a vertex $r$ that is contained in at least one, but not all, optimum vertex covers of $G$, and is not adjacent to all other vertices of $G$, i.e., $N[r] \subsetneq V(G)$. Let $Y$ be a minimal blocking set of $G$ with $r \notin Y$.*[2]

[1] Such a blocking set exists because either $V(G) \setminus \{r\}$ is a blocking set of $G$ or $\text{VC}(G) = |V(G)| - 1$, and because $\text{VC}(G) = |V(G)| - 1$ implies that $\text{VC}(G - r) < \text{VC}(G)$.

[2] Such a blocking set exists because $V(G) \setminus \{r\}$ is a blocking set of $G$; otherwise every optimum solution contains all except one vertex of $G$ which implies that $V(G)$ is the only (minimal) blocking set of $G$. But the only graphs that have the entire set of vertices as a minimal blocking set are cliques.

(i) *The set $Y$ is also a blocking set of $G - r$ and the set $Y \setminus N[r]$ is a blocking set of $G - N[r]$.*

(ii) *Let $Y'$ be a minimal blocking set of $G - r$, and let $\widetilde{Y}$ be a minimal blocking set of $G - N[r]$. The set $Y' \cup \widetilde{Y}$ is a blocking set of $G$.*

(iii) *For all vertices $y \in Y$ it holds that $Y \setminus \{y\}$ is not a blocking set of $G - r$ or that $Y \setminus (\{y\} \cup N[r])$ is not a blocking set of $G - N[r]$.*

**Proof.** Since $G$ contains an optimum vertex cover that contains $r$ and an optimum vertex cover that contains $N(r)$ it follows from Lemma 3.6 item (ii) that $Y \setminus \{r\} = Y$ is a blocking set of $G - r$ and that $Y \setminus N(r) = Y \setminus N[r]$ is a blocking set of $G - N(r)$. Furthermore, it holds that $\text{VC}(G - r) = \text{VC}(G) - 1$ and that $\text{VC}(G - N(r)) = \text{VC}(G) - |N(r)|$. Note that $Y \setminus N[r]$ is also a blocking set of $G - N[r]$ and that $\text{VC}(G - N[r]) = \text{VC}(G) - |N(r)|$ because vertex $r$ is isolated in $G - N(r)$ and not contained in $Y \setminus N(r)$. This proves item (i).

Next, we prove item (ii). Let $Y'$ be a minimal blocking set of $G - r$ and let $\widetilde{Y}$ be a minimal blocking set of $G - N[r]$. Assume for contradiction that $\overline{Y} = Y' \cup \widetilde{Y}$ is not a blocking set of $G$. Thus, there exists an optimum vertex cover $X$ of $G$ that contains $\overline{Y}$. If the vertex $r$ is contained in $X$, then $X \setminus \{r\}$ is an optimum vertex cover of $G - r$ that contains $\overline{Y}$ and therefore also $Y'$. This contradicts the assumption that $Y'$ is a blocking set of $G - r$. If the vertex $r$ is not contained in $X$, then $X$ contains the vertex set $N(r)$ and $X \setminus N(r)$ is an optimum vertex cover of $G - N[r]$ that contains $\widetilde{Y} \subseteq \overline{Y} \setminus N(r)$. But this is a contradiction to the assumption that $\widetilde{Y}$ is a blocking set of $G - N[r]$. Overall, this proves that $Y' \cup \widetilde{Y}$ is a blocking set of $G$.

Finally, we show item (iii). Assume for contradiction that there exists a vertex $y \in Y$ such that $Y \setminus \{y\}$ is a blocking set in $G - r$ and such that $Y \setminus (\{y\} \cup N[r])$ is a blocking set in $G - N[r]$. Let $Y' \subseteq Y \setminus \{y\}$ be a minimal blocking set in $G - r$ and let $\widetilde{Y} \subseteq Y \setminus (\{y\} \cup N[r])$ be a minimal blocking set in $G - N[r]$. It follows from item (ii) that $Y' \cup \widetilde{Y} \subseteq Y \setminus \{y\} \subsetneq Y$ is a blocking set of $G$. This contradicts the assumption that $Y$ is a minimal blocking set of $G$ and concludes the proof of item (iii). ■

Using the above lemma we are able to prove the last lemma we need to prove Theorem 5.17. Observe that we can apply the following Lemma in our proof of Theorem 5.17 only when our class $\mathcal{C}$ is hereditary, because we need to bound the size of minimal blocking sets of connected graphs where we delete more than the root of an optimum elimination forest. These vertices could be part of the base components of an optimum elimination forest which may increase the elimination distance because the base components are no longer contained in class $\mathcal{C}$ when $\mathcal{C}$ is not hereditary.

**Lemma 5.22.** *Let $G$ be a graph that contains a vertex $r$ that is contained in at least one, but not all, optimum vertex covers of $G$, and that is not adjacent to all other vertices of $G$; hence $\text{VC}(G - r) + 1 = \text{VC}(G)$ and $N[r] \subsetneq V(G)$. Let $Y$ be a minimal blocking set of $G$ with $r \notin Y$.*[3] *At least one of the following three cases holds:*

[3]The existence follows from Lemma 5.21.

1. *The set $Y$ is also a minimal blocking set of $G - r$.*
2. *The set $Y = Y \setminus N(r)$ is also a minimal blocking set of $G - N[r]$.*
3. *There exists a minimal blocking set $Y' \subsetneq Y$ of $G - r$, and the set $\hat{Y} = Y \setminus Y'$ is a minimal blocking set of $G - Y'$. Furthermore, $\hat{Y}$ is not a minimal blocking set of $G - Y' - r$, however, there exists a non-empty set $Z \subseteq N(r) \setminus Y'$ such that $\hat{Y} \cup Z$ is a minimal blocking set of $G - Y' - r$.*

**Proof.** It follows from Lemma 5.21 item (i) that $Y$ is a blocking set of $G - r$ and that $Y \setminus N[r]$ is a blocking set of $G - N[r]$. If $Y$ is also a minimal blocking set of $G - r$ then Case 1 holds, and if $Y \setminus N[r] = Y$ is a minimal blocking set of $G - N[r]$ then Case 2 holds. Thus, in the following we assume that neither Case 1 nor Case 2 applies.

Let $Y' \subsetneq Y$ be a minimal blocking set of $G - r$ and let $\widetilde{Y} \subseteq Y \setminus N(r)$ be a minimal blocking set of $G - N[r]$. Consider the graph $G - Y'$ and the set $\hat{Y} = Y \setminus Y'$. Since $Y' \subsetneq Y$ and since $Y$ is a minimal blocking set of $G$, there exists an optimum vertex cover $X$ of $G$ that contains $Y'$. Therefore, it follows from Lemma 3.6 item (i) that $\textsc{vc}(G - Y') + |Y'| = \textsc{vc}(G)$ and from items (ii), (iii) that $\hat{Y}$ is a minimal blocking set of $G - Y'$.

It remains to show the second part, namely, that $\hat{Y}$ is not a minimal blocking set of $G - r - Y'$. First, we show that $\textsc{vc}(G - Y') = \textsc{vc}(G - r - Y')$. Let $\hat{X}$ be an optimum vertex cover of $G - r - Y'$. It holds that $\hat{X} \cup Y'$ is a vertex cover of $G - r$. Since $\hat{X} \cup Y'$ is a vertex cover of $G - r$ that contains $Y'$, and $Y'$ is a blocking set of $G - r$ it holds that $|\hat{X}| + |Y'| \geq \textsc{vc}(G - r) + 1$. Thus, the optimum vertex $\hat{X}$ of $G - r - Y'$ has size at least $\textsc{vc}(G - r) + 1 - |Y'| = \textsc{vc}(G) - |Y'| = \textsc{vc}(G - Y')$. This concludes the proof that $\textsc{vc}(G - Y') = \textsc{vc}(G - r - Y')$ because it always holds that $\textsc{vc}(G - Y') \geq \textsc{vc}(G - r - Y')$.

Now, we are able to prove that $\hat{Y}$ is not a blocking set of $G - r - Y'$. Let $y \in Y' \setminus \widetilde{Y}$. Note that $Y' \setminus \widetilde{Y}$ is not empty because $Y' \cup \widetilde{Y} = Y$ (Lemma 5.21 item (ii)) and because $\widetilde{Y} \subsetneq Y$. It holds that the set $Y \setminus \{y\}$ is not a blocking set of $G - r$ since $Y \setminus \{y\} \supseteq \widetilde{Y}$ is a blocking set of $G - N[r]$ and since Lemma 5.21 item (iii) states that $Y \setminus \{y\}$ cannot be a blocking set in $G - r$ and $G - N[r]$. Thus, there exists an optimum vertex cover $X'$ of $G - r$ with $\hat{Y} \subseteq Y \setminus \{y\} \subseteq X'$. The set $X = X' \cup \{r\}$ is an optimum vertex cover of $G$ because $\textsc{vc}(G) = \textsc{vc}(G - r) + 1$. Furthermore, the set $X$ contains the vertex set $\hat{Y}$ as well as the vertex $r$, but not the vertex $y$ (because $Y$ is a blocking set of $G$ and $Y \setminus \{y\} \subseteq X$). Let $\hat{X} = X \setminus Y'$. It holds that $\hat{X}$ is a vertex cover of $G - Y'$ that contains the set $\hat{Y}$ and vertex $r$. Since $Y' \setminus \{y\} \subseteq X$, the set $\hat{X}$ has size $\textsc{vc}(G) - |Y'| + 1 = \textsc{vc}(G - Y') + 1$. Thus, the set $\hat{X} \setminus \{r\}$ is an optimum vertex cover of $G - r - Y'$ because $\textsc{vc}(G - Y') = \textsc{vc}(G - r - Y')$. Now, the optimum vertex cover $\hat{X} \setminus \{r\}$ of $G - r - Y'$ contains the set $\hat{Y}$ which implies that $\hat{Y}$ is not a blocking set of $G - r - Y'$.

Observe that the graph $G - Y'$ together with the vertex $r$ and the set $\hat{Y}$ fulfills the properties of Lemma 5.19. Hence, there exists a set $Z \subseteq N_{G-Y'}(r) = N(r) \setminus Y'$ such that $\hat{Y} \cup Z$ is a minimal blocking set of $G - Y' - r$. Since, $\hat{Y}$ is not a blocking set in $G - Y' - r$ it holds that $Z \neq \emptyset$. This concludes the proof. ∎

Combining Lemma 5.18 to Lemma 5.22 we can now prove Theorem 5.17.

**Proof of Theorem 5.17.** We prove Theorem 5.17 by induction over the integer $d$. In the base case assume that $d = 0$. Since every graph $G$ with $\mathrm{ed}_{\mathcal{C}}(G) = 0$ is contained in graph class $\mathcal{C}$ it follows that $\beta_{\mathcal{C}}(0) = \max\{\beta(G) \mid \mathrm{ed}_{\mathcal{C}}(G) = 0\} \leq \beta_{\mathcal{C}}$. Hence, the base case holds for $\beta_{\mathcal{C}} = 1$. For $\beta_{\mathcal{C}} \geq 2$ it holds that $(\beta_{\mathcal{C}} - 1) \cdot 2^d + 1 = \beta_{\mathcal{C}} \geq \beta(G)$. Thus, the base case holds for all graph classes where $\beta_{\mathcal{C}}$ is bounded.

For the induction step, we assume that the statement is true for all integers less than $d$, and all robust hereditary graph classes $\mathcal{C}$ where $\beta_{\mathcal{C}}$ is bounded.

Let $\mathcal{C}$ be a robust hereditary graph class where $\beta_{\mathcal{C}}$ is bounded, let $G$ be any graph with $\mathrm{ed}_{\mathcal{C}}(G) = d$, and let $Y$ be a minimal blocking set of $G$. We can assume that graph $G$ is connected because $\mathcal{C}$ is robust which implies that $\mathrm{ed}_{\mathcal{C}}(G) = \max\{\mathrm{ed}_{\mathcal{C}}(G') \mid G' \text{ connected component of } G\}$ and because $Y$ is contained in at most one connected component of $G$ (Proposition 3.5 item (iii)) which implies that $\beta(G) = \max\{\beta(G') \mid G' \text{ connected component of } G\}$. Thus, to bound $\beta_{\mathcal{C}}(d)$ it is enough to bound $\beta(G)$ for every connected graph $G$ with $\mathrm{ed}_{\mathcal{C}}(G) \leq d$. Let $r$ be the root of an optimum elimination forest of $G$ to graph class $\mathcal{C}$. (There exists exactly one root because $G$ is connected.) We will show that $Y$ has the requested size by distinguishing between five cases:

**Case 1:** Assume that $\mathrm{VC}(G) = \mathrm{VC}(G - r)$ and that $r \in Y$.

It follows from Lemma 5.18 item (i) that $Y = \{r\}$. Since $2^{d-1} + 1 \geq 1$ and $(\beta_{\mathcal{C}} - 1) \cdot 2^d + 1 \geq 1$ for $\beta_{\mathcal{C}} \geq 2$, the set $Y$ has the required size.

**Case 2:** Assume that $\mathrm{VC}(G) = \mathrm{VC}(G - r) + 1$ and that $r \in Y$.

Observe that the graph $G$ and the set $Y$ fulfill the requirements of Lemma 5.18 item (ii). Thus, the set $Y \setminus \{r\}$ is a minimal blocking set of $G - r$. Since $r$ was the root of the elimination forest of $G$ it holds that $\mathrm{ed}_{\mathcal{C}}(G - r) \leq d - 1$. Hence, we can bound the size of $Y$ by $\beta(G - r) + 1 \leq \beta_{\mathcal{C}}(d - 1) + 1$.

**Case 3:** Assume that $\mathrm{VC}(G) = \mathrm{VC}(G - r)$ and that $r \notin Y$.

The graph $G$ together with the set $Y$ and the vertex $r$ fulfills the conditions of Lemma 5.19. This implies that there exists a (possibly empty) set $Z \subseteq N(r)$ such that $Y \cup Z$ is a minimal blocking set of $G - r$. Since $G - r$ has elimination distance at most $d - 1$ to graph class $\mathcal{C}$ we can bound the size of $Y$ by $\beta(G - r) \leq \beta_{\mathcal{C}}(d - 1)$.

**Case 4:** Assume that $\mathrm{VC}(G) = \mathrm{VC}(G - r) + 1$, that vertex $r$ is contained in every optimum vertex cover of $G$, and that $r \notin Y$.

Note that graph $G$, vertex $r$ and the minimal blocking set $Y$ fulfill the requirements of Lemma 5.20. Therefore, the set $Y$ is also a minimal blocking set of $G - r$. This implies that $|Y| \leq \beta_{\mathcal{C}}(d - 1)$ because $G - r$ has elimination distance at most $d - 1$ to graph class $\mathcal{C}$.

**Case 5:** Assume that $\text{VC}(G) = \text{VC}(G-r)+1$, that vertex $r$ is contained in at least one, but not all, optimum vertex covers of $G$, and that $r \notin Y$.

Observe that $N[r] \subsetneq V(G)$ because there exists a solution that does not contain vertex $r$ which implies that this solution contains $N(r)$. This in turn would imply that if $N[r] = V(G)$ then $Y \subseteq V(G) \setminus \{r\} = N(r)$ is not a blocking set of $G$ which would contradict the assumption.

This is the only case where we need that the class $\mathcal{C}$ is hereditary, because we have to bound the size of a minimal blocking set in a subgraph of $G - r$. The graph $G$, the vertex $r$ and the set $Y$ fulfill the conditions of Lemma 5.22. If Case 1 of Lemma 5.22 holds then $Y$ is a minimal blocking set of $G - r$ and it follows that $|Y| \le \beta(G - r) \le \beta_{\mathcal{C}}(d-1)$.

Next, assume that Case 2 of Lemma 5.22 holds. Since graph class $\mathcal{C}$ is hereditary and deleting vertices of the elimination forest of $G$ can only decrease the elimination distance, it follows that $\text{ed}_{\mathcal{C}}(G-N[r]) \le d-1$. Thus, we can bound the size of the set $Y$, which is a minimal blocking set of $G-N[r]$, by $\beta(G - N[r]) \le \beta_{\mathcal{C}}(d-1)$ (Observation 5.14).

Now, assume that Case 3 of Lemma 5.22 holds. Recall that $Y = Y' \dot\cup \hat{Y}$ where $Y'$ is a minimal blocking set in $G-r$ and $\hat{Y} = Y \setminus Y'$ is a minimal blocking set in $G-Y'$. Thus, we can bound the size of $Y'$ by $\beta(G - r) \le \beta_{\mathcal{C}}(d-1)$. The size of $\hat{Y}$ is at most $\beta(G - Y' - r) - 1$ because there exists a non-empty set $Z \subseteq N(r) \setminus Y'$ such that $\hat{Y} \cup Z$ is a minimal blocking set of $G - Y' - r$. Since $G - Y' - r$ is a subgraph of $G - r$ and the graph class $\mathcal{C}$ is hereditary it holds that $\text{ed}_{\mathcal{C}}(G' - Y' - r) \le d-1$. This implies that $|\hat{Y}| \le \beta(G - Y' - r) - 1 \le \beta_{\mathcal{C}}(d-1) - 1$ (Observation 5.14). Combining the bound for $Y'$ and $\hat{Y}$ we obtain that $|Y| \le 2\beta_{\mathcal{C}}(d-1) - 1$.

Overall, we showed that $|Y| \le \max\{1, \beta_{\mathcal{C}}(d-1)+1, 2\beta_{\mathcal{C}}(d-1)-1\}$. For $d = 1$ and $\beta_{\mathcal{C}} = 1$ this implies that $|Y| \le 2 = 2^{1-1}+1$ because $\beta_{\mathcal{C}}(0) = \beta_{\mathcal{C}} = 1$. Thus, $\beta_{\mathcal{C}}(1) \le 2^{1-1}+1$ because we picked an arbitrary graph $G$ with $\text{ed}_{\mathcal{C}}(G) = 1$, and an arbitrary minimal blocking set of $G$.

If $d \ge 2$ or $\beta_{\mathcal{C}} \ge 2$, then it holds that $\max\{1, \beta_{\mathcal{C}}(d-1)+1, 2\beta_{\mathcal{C}}(d-1)-1\} = 2\beta_{\mathcal{C}}(d-1)-1$ because $\beta_{\mathcal{C}}(d-1) \ge 2$. Hence, we can bound the size of the set $Y$ by $2\beta_{\mathcal{C}}(d-1)-1$ when $d \ge 2$ or $\beta_{\mathcal{C}} \ge 2$. Furthermore, we can bound $\beta_{\mathcal{C}}(d)$ by $2\beta_{\mathcal{C}}(d-1)-1$ when $d \ge 2$ or $\beta_{\mathcal{C}} \ge 2$ because we picked an arbitrary graph $G$ with $\text{ed}_{\mathcal{C}}(G) \le d$ as well as an arbitrary minimal blocking set of $G$.

For $\beta_{\mathcal{C}} = 1$ and $d \ge 2$ this leads to the following upper bound for $\beta_{\mathcal{C}}(d)$:

$$\beta_{\mathcal{C}}(d) \le 2 \cdot \beta_{\mathcal{C}}(d-1) \le 2 \cdot \left(2^{(d-1)-1}+1\right) - 1 = 2^{d-1}+1$$

The second inequality follows from the induction hypothesis. Finally, we show the upper bound for $\beta_{\mathcal{C}}(d)$ for the case that $\beta_{\mathcal{C}} \ge 2$ (and $d \ge 1$):

$$\beta_{\mathcal{C}}(d) \le 2 \cdot \beta_{\mathcal{C}}(d-1) \le 2 \cdot \left((\beta_{\mathcal{C}}-1)2^{(d-1)}+1\right) - 1 = (\beta_{\mathcal{C}}-1)2^{d}+1$$

Again, the second inequality follows from the induction hypothesis. This concludes the proof. ■

It follows from Theorem 5.12 and Theorem 5.17 that the bound for $\beta_{\mathcal{C}}(d)$ is tight for all hereditary graph classes $\mathcal{C}$, proving Theorem 5.1.

**Theorem 5.1.** *Let $\mathcal{C}$ be a robust hereditary graph class where $\beta_{\mathcal{C}}$ is bounded. For every integer $d \geq 1$ it holds that*

$$\beta_{\mathcal{C}}(d) = \begin{cases} 2^{d-1} + 1 & \text{, if } \beta_{\mathcal{C}} = 1, \\ (\beta_{\mathcal{C}} - 1)2^d + 1 & \text{, if } \beta_{\mathcal{C}} \geq 2. \end{cases}$$

**Non-hereditary graph classes.** In the remaining part of this section, we will show that we can obtain an upper bound for $\beta_{\mathcal{C}}(d)$ even if $\mathcal{C}$ is not hereditary, but fulfills some other additional properties. Nevertheless, this leads to a weaker upper bound.

**Definition 5.23.** We say that a graph class $\mathcal{C}$ is *$f$-solid*, if $\beta_{\mathcal{C}+c} \leq f(c) = f(\beta_{\mathcal{C}}, c)$ for a computable function $f$.

As mentioned earlier, the requirement that graph class $\mathcal{C}$ is $f$-solid is not unnatural: Any graph that has a $\mathcal{C}$-modulator of size at most $d$ has elimination distance at most $d$ to class $\mathcal{C}$. Thus, if there exists an upper bound for $\beta_{\mathcal{C}}(d)$ then there exists an upper bound for the minimal blocking set size of graphs that have a $\mathcal{C}$-modulator of size at most $d$. Assume that the minimal blocking set size of a graph $G$ that has a $\mathcal{C}$-modulator of size at most $c$ is bounded by $f(c)$, and let $G$ be a graph that is contained in class $\mathcal{C} + c$. Since every connected component $H$ of graph $G$ has a $\mathcal{C}$-modulator of size at most $c$, it follows that $\beta(H) \leq f(c)$ for all connected components $H$ of $G$. Together with the fact that $\beta(G) = \max\{\beta(H) \mid H \text{ connected component of } G\}$ it follows that $\beta_{\mathcal{C}+c} \leq f(c)$.

Observation 5.24. It holds that $\beta_{\mathcal{C}} < \beta_{\mathcal{C}+1}$ for all classes $\mathcal{C}$ (Observation 5.14). This implies that $\beta_{\mathcal{C}+c} \geq 2$ for all $c \geq 1$.

**Theorem 5.25.** *Let $\mathcal{C}$ be an $f$-solid and robust graph class where $\beta_{\mathcal{C}}$ is bounded, and let $d \geq 0$. It holds that*

$$\beta_{\mathcal{C}+c}(d) \leq \left( \sum_{i=0}^{d} \binom{d}{i} f(c+i) \right) - 2^d + 1.$$

To prove Theorem 5.25 we need an additional lemma that allows us to bound the size of a minimal blocking set of subgraphs of $G - r$, where $G$ is a connected graph with $\mathrm{ed}_{\mathcal{C}}(G) \leq d$ and where $r$ is the root of an optimum elimination forest of $G$ to graph class $\mathcal{C}$.

**Lemma 5.26.** *Let $\mathcal{C}$ be a robust graph class, let $G$ be a connected graph with $\mathrm{ed}_{\mathcal{C}+c}(G) = d$, let $r$ be the root of an optimum elimination forest $F$ of $G$ to graph class $\mathcal{C}+c$ with $c \geq 0$, and let $Z \subseteq V(G)$ such that $\mathrm{VC}(G-Z)+|Z| = \mathrm{VC}(G)$. It holds that $\beta(G-Z-r) \leq \beta_{\mathcal{C}+c+1}(d-1)$.*

**Proof.** We construct a graph $\hat{G}$ with $\mathrm{ed}_{\mathcal{C}+c+1}(\hat{G}) = d-1$ that has the property that for every minimal blocking set $Y$ of $G-Z-r$ there exists a (possibly empty) set $\hat{Z} \subseteq Z$ such that $Y \cup \hat{Z}$ is a minimal blocking set of $\hat{G}$.

To construct graph $\hat{G}$ from graph $G$ we first delete all vertices from $Z$ that are part of the elimination forest $F$ of $G$ to graph class $\mathcal{C}+c$. Let $Z' \subseteq Z$ be the vertices of $Z$ that are contained in the base components of the elimination forest $F$ of $G$ to graph class $\mathcal{C}+c$. Observe that $\mathrm{ed}_{\mathcal{C}+c}(G - Z \setminus Z') \leq d$ because we only delete vertices of the elimination forest $F$. Let $\mathcal{H}$ be the set of base components of the elimination forest $F$ of $G$ to graph class $\mathcal{C}+c$. We can assume, without loss of generality, that each base component of the elimination forest of $G$ to graph class $\mathcal{C}+c$ is connected because $\mathcal{C}$, and therefore, also $\mathcal{C}+c$ are robust. Thus, we can split every disconnected base component of an elimination forest to graph class $\mathcal{C}+c$ into its connected components without changing the height of the elimination forest.

For base component $H \in \mathcal{H}$ with $V(H) \cap Z' \neq \emptyset$ we add a vertex $v_H$ to graph $H$ and connect it to all vertices in $Z' \cap V(H)$. The resulting graph is $\widetilde{G}$. Note that we add at most one vertex to each base component in $\mathcal{H}$, and that every graph in $\mathcal{H}$ is connected. Thus, every base component in $\mathcal{H}$ now belongs to the class $\mathcal{C}+c+1$ which implies that $\mathrm{ed}_{\mathcal{C}+c+1}(\widetilde{G}) \leq d$ (because we can use the same elimination forest as for $G - Z \setminus Z'$). Finally, we delete vertex $r$ from $\widetilde{G}$ to obtain $\hat{G}$. It holds that $\mathrm{ed}_{\mathcal{C}+c+1}(\hat{G}) \leq d-1$, because $r$ is the root of an optimum elimination forest of $\widetilde{G}$ to graph class $\mathcal{C}+c+1$.

First, we show that $\mathrm{VC}(\hat{G}) = \mathrm{VC}(G-Z-r)+|Z'|$. Let $X$ be an optimum vertex cover of $G-Z-r$. The set $X \cup Z'$ is a vertex cover of $\hat{G}$ because $G-(Z \setminus Z')-r$ is a subgraph of $\hat{G}$ and because every newly added vertex is only adjacent to vertices in $Z'$. Hence, $\mathrm{VC}(\hat{G}) \leq \mathrm{VC}(G-Z-r)+|Z'|$.

For the other direction, let $\hat{X}$ be an optimum vertex cover of $\hat{G}$. If $Z' \subseteq \hat{X}$ then $\hat{X} \setminus Z'$ is a vertex cover of $G-Z-r$ of size $\mathrm{VC}(\hat{G})-|Z'|$. Note that $\hat{X}$ does not contain any newly added vertex in this case because these vertices are only adjacent to vertices in $Z'$. If $\hat{X}$ contains a newly added vertex then we add $r$ as well as $Z \setminus Z'$ to $\hat{X}$ and delete all newly added vertices from $\hat{X}$. The resulting set is a vertex cover of $G$ of size at most $\mathrm{VC}(\hat{G})+|Z \setminus Z'|$ because $G-(Z \setminus Z')-r$ is a subgraph of $\hat{G}$ and because the set $\hat{X}$ contains at least one newly added vertex. Together with the assumption that $\mathrm{VC}(G) = \mathrm{VC}(G-Z)+|Z| \geq \mathrm{VC}(G-Z-r)+|Z|$ it follows that $\mathrm{VC}(G-Z-r) \leq \mathrm{VC}(G)-|Z| \leq \mathrm{VC}(\hat{G})+|Z \setminus Z'|-|Z| = \mathrm{VC}(\hat{G})-|Z'|$. Overall, we showed that $\mathrm{VC}(\hat{G}) = \mathrm{VC}(G-Z-r)+|Z'|$.

Let $Y$ be a minimal blocking set of $G-Z-r$. We will show that $Y \cup Z'$ is a blocking set of $\hat{G}$. Assume for contradiction that $Y \cup Z'$ is not a blocking set of $\hat{G}$, and let $\hat{X}$ be an optimum vertex cover of $\hat{G}$ that contains $Y \cup Z'$. This implies that $\hat{X} \setminus Z'$ is an optimum vertex cover of $G-Z-r$ that contains $Y$ because $G-Z-r$ is a subgraph of

$\hat{G}$, because $Z' \subseteq \hat{X}$, and because $\mathrm{VC}(\hat{G}) = \mathrm{VC}(G - Z - r) + |Z'|$. This contradicts the assumption that $Y$ is a (minimal) blocking set of $G - Z - r$ and proves that $Y \cup Z'$ is a blocking set of $\hat{G}$.

Now, let $\hat{Y} \subseteq Y \cup Z'$ be a minimal blocking set of $\hat{G}$. We will show that $Y \subseteq \hat{Y}$. This directly implies that $\beta(G - Z - r) \leq \beta(\hat{G}) \leq \beta_{\mathcal{C}+c+1}(d-1)$. Observe that $Z'$ is not a blocking set of $\hat{G}$ because there exists an optimum vertex cover of $\hat{G}$ that contains $Z'$, namely every optimum vertex cover of $G - Z - r$ together with the set $Z'$; hence $\hat{Y} \setminus Z' \neq \emptyset$. If $\hat{Y} \setminus Z' \subsetneq Y$ then $\hat{Y} \setminus Z'$ is not a blocking set of $G - Z - r$ because $Y$ is a minimal blocking set of $G - Z - r$. Thus, there exists an optimum vertex cover $X$ of $G - Z - r$ which contains the set $\hat{Y} \setminus Z' \subsetneq Y$. But, $\hat{X} = X \cup Z'$ is a vertex cover of $\hat{G}$, because $G - (Z \setminus Z') - r$ is a subgraph of $\hat{G}$ and every newly added vertex is only adjacent to vertices in $Z'$. Furthermore, $\hat{X}$ is an optimum vertex cover of $\hat{G}$ because $|\hat{X}| = |X| + |Z'| = \mathrm{VC}(G - Z - r) + |Z'| = \mathrm{VC}(\hat{G})$, and it holds that $\hat{Y} \subseteq \hat{X}$. This contradicts the assumption that $\hat{Y}$ is a (minimal) blocking set of $\hat{G}$, and concludes the proof. ■

Now, we are able to prove Theorem 5.17 using the above lemma.

**Proof of Theorem 5.25.** The proof of Theorem 5.25 is very similar to the proof of Theorem 5.17. We prove Theorem 5.25 also by induction over the integer $d$, and for every integer $d$ we show that for all graphs $G$ with $\mathrm{ed}_{\mathcal{C}+c}(G) \leq d$, and every minimal blocking set $Y$ of $G$, we can bound the size of $Y$ by the claimed upper bound. Furthermore, we use the same definition by cases as in the proof of Theorem 5.25.

In the base case assume that $d = 0$. Every graph $G$ with $\mathrm{ed}_{\mathcal{C}+c}(G) = 0$ is a graph of the class $\mathcal{C} + c$. Therefore, it holds that $\beta(G) \leq \beta_{\mathcal{C}+c} = f(c) = \sum_{i=0}^{0} \binom{0}{i} f(c+i) - 2^0 + 1$.

For the inductive step we assume that the statement holds for all integers less than $d$, and for all $f$-solid, robust graph classes $\mathcal{C} + c$ with $c \geq 0$.

Let $\mathcal{C}$ be any $f$-solid, robust graph class where $\beta_{\mathcal{C}}$ is bounded, let $G$ be any graph with $\mathrm{ed}_{\mathcal{C}+c}(G) \leq d$, and let $Y$ be a minimal blocking set of $G$. As in the proof of Theorem 5.17 we can assume that the graph $G$ is connected. Let $r$ be the root of an optimum elimination forest of $G$ to graph class $\mathcal{C} + c$. We show that $Y$ has the desired size by distinguishing between the same five cases as in Theorem 5.17:

Observe that Cases 1 to 4 of the proof of Theorem 5.17 never used that the graph class $\mathcal{C}$ is hereditary. We only used that we can bound $\beta(G - r)$ using our induction hypothesis. Thus, even when graph class $\mathcal{C} + c$ is not hereditary, we obtain the same bound for the size of a minimal blocking set $Y$ in these cases, because $\mathrm{ed}_{\mathcal{C}+c}(G - r) \leq d - 1$. Recall that Case 2 leads to the worst upper bound for the size of $Y$, namely $|Y| \leq \beta(G - r) + 1 \leq \beta_{\mathcal{C}+c}(d-1) + 1$.

Now, we consider the remaining case where we assumed that the graph class $\mathcal{C}$ is hereditary.

**Case 5:** Assume that $\mathrm{VC}(G) = \mathrm{VC}(G - r) + 1$, that vertex $r$ is contained in at least one, but not all, optimum vertex covers of $G$, and that $r \notin Y$.

Observe, the graph $G$, the vertex $r$, and the set $Y$ fulfill the requirements of Lemma 5.22.[4] As in the proof of Theorem 5.17 we distinguish between the three cases of Lemma 5.22. If Case 1 of Lemma 5.22 holds then $Y$ is also a minimal blocking set of $G - r$ which implies that $|Y| \leq \beta(G - r) \leq \beta_{\mathcal{C}+c}(d-1)$. Next, assume that Case 2 of Lemma 5.22 holds. Thus, the set $Y = Y \setminus N[r]$ is also a minimal blocking set of $G - N[r]$. Since there exists an optimum vertex cover of $G$ that contains the set $N(r)$ it follows from Lemma 5.26 that $|Y| \leq \beta(G - N[r]) \leq \beta_{\mathcal{C}+c+1}(d-1)$.

Finally, assume that Case 3 (and neither Case 1 or Case 2) of Lemma 5.22 holds. As in the proof of Case 5 of Theorem 5.25 we bound the size of the minimal blocking set $Y' \subsetneq Y$ of $G - r$ and the minimal blocking set $\hat{Y} = Y \setminus Y'$ of $G - Y'$. Obviously, the size of $Y'$ is at most $\beta_{\mathcal{C}+c}(d-1)$ because $\mathrm{ed}_{\mathcal{C}+c}(G - r) \leq d - 1$. Recall, the size of $\hat{Y}$ is at most $\beta(G - Y' - r) - 1$ because $\hat{Y}$ is not a blocking set of $G - Y' - r$ whereas $\hat{Y} \cup Z \subseteq \hat{Y} \cup (N(r) \setminus Y')$ is a minimal blocking set of $G - Y' - r$ for a set $Z' \subseteq N(r) \setminus Y'$. Since $Y' \subsetneq Y$ and since $Y$ is a minimal blocking set of $G$, there exists an optimum vertex cover of $G$ that contains the set $Y'$. Thus, we can use Lemma 5.26 to bound the size of $\beta(G - Y' - r)$ by $\beta_{\mathcal{C}+c+1}(d-1)$. Hence, it holds that $|\hat{Y}| \leq \beta(G - Y' - r) - 1 \leq \beta_{\mathcal{C}+c+1}(d-1) - 1$. We obtain that $|Y| \leq \beta_{\mathcal{C}+c}(d-1) + \beta_{\mathcal{C}+c+1}(d-1) - 1$.

Overall, we can bound the size of $Y$ by $\beta_{\mathcal{C}+c}(d-1) + \beta_{\mathcal{C}+c+1}(d-1) - 1$, because $\beta_{\mathcal{C}+c'}(d) \geq 2$ when $c' \geq 1$ (see Observation 5.24). Together with the induction hypothesis we can bound the size of $Y$ as follows:

$$
\begin{aligned}
|Y| &\leq \beta_{\mathcal{C}+c}(d-1) + \beta_{\mathcal{C}+c+1}(d-1) - 1 \\
&\overset{(IH)}{\leq} \left(\sum_{i=0}^{d-1} \binom{d-1}{i} f(c+i) - 2^{d-1} + 1\right) + \left(\sum_{i=0}^{d-1} \binom{d-1}{i} f(c+1+i) - 2^{d-1} + 1\right) - 1 \\
&= \left(\sum_{i=0}^{d-1} \binom{d-1}{i} f(c+i)\right) + \left(\sum_{i=1}^{d} \binom{d-1}{i-1} f(c+i)\right) - 2 \cdot 2^{d-1} + 2 - 1 \\
&= \binom{d-1}{0} f(c) + \sum_{i=1}^{d-1} \left(\binom{d-1}{i} + \binom{d-1}{i-1}\right) f(c+i) + \binom{d-1}{d-1} f(c+d) - 2^d + 1 \\
&= \binom{d}{0} f(c) + \sum_{i=1}^{d-1} \binom{d}{i} f(c+i) + \binom{d}{d} f(c+d) - 2^d + 1 \\
&= \left(\sum_{i=0}^{d} \binom{d}{i} f(c+i)\right) - 2^d + 1
\end{aligned}
$$

This concludes the proof. ■

[4] It holds that $N[r] \subsetneq V(G)$. Otherwise, it would hold that $Y$ is not a blocking set of $G$ (see proof of Theorem 5.17 Case 5.)

Recall, $\mathcal{C}_{\text{LP}} = \{G \text{ graph} \mid \text{VC}(G) = \text{LP}(G)\}$ is the set of graphs where the size of an optimum vertex cover equals the value of an optimum LP solution.[5] It holds that every graph $G \in \mathcal{C}_{\text{LP}} + c$ is a $c$-quasi-integral graph: Let $H$ be a connected component of $G$, and let $X_H \subseteq V(H)$ of size at most $c$ such that $H - X_H \in \mathcal{C}_{\text{LP}}$ (exists by definition of $\mathcal{C} + c$). It holds that $\text{VC}(H - X_H) + |X_H| \geq \text{VC}(H)$, that $\text{LP}(H - X_H) \leq \text{LP}(H)$, and that $\text{VC}(H - X_H) = \text{LP}(H - X_H)$ because $H - X_H \in \mathcal{C}_{\text{LP}}$. This implies that $\text{VC}(H) - \text{LP}(H) \leq |X_H| \leq c$ and shows that $G$ is $c$-quasi-integral. Thus, it follows from Theorem 5.7 that $\mathcal{C}_{\text{LP}}$ is a non-hereditary graph class that is $f$-solid with $f(c) = f(\beta_{\mathcal{C}}, c) = 2c + \beta_{\mathcal{C}_{\text{LP}}} = 2c + 2$ because $\mathcal{C} + c \subseteq \mathcal{C}_{c\text{-qi}}$.

**Corollary 5.27.** *It holds that* $\beta_{\mathcal{C}_{\text{LP}}}(d) \leq (\beta_{\mathcal{C}_{\text{LP}}} + d - 1) \cdot 2^d + 1 = (d+1) \cdot 2^d + 1$.

**Proof.** This follows directly from Theorem 5.25, because $\mathcal{C}_{\text{LP}}$ is a $(2c + \beta_{\mathcal{C}_{\text{LP}}})$-solid graph class, where $\beta_{\mathcal{C}_{\text{LP}}} = 2$. This implies that

$$\begin{aligned}\beta_{\mathcal{C}_{\text{LP}}}(d) &\leq \sum_{i=0}^{d} \binom{d}{i} f(\beta_{\mathcal{C}_{\text{LP}}}, i) - 2^d + 1 = \sum_{i=0}^{d} \binom{d}{i} (2 \cdot i + 2) - 2^d + 1 \\ &= (d+1) \cdot 2^d + 1\end{aligned}$$

■

## 5.4. Summary

In this chapter we showed for some robust graph classes $\mathcal{C}$, that have bounded minimal blocking set size, that variations of $\mathcal{C}$ also have bounded minimal blocking set size. We started by giving tight bounds for the minimal blocking set size of the classes $\mathcal{C}_{d\text{-qf}}$, $\mathcal{C}_{d\text{-qb}}$, $\mathcal{C}_{d\text{-qi}}$ and $\mathcal{C}_{2\text{LP}-\text{MM}}$. Using half-integral optimum solutions to $\text{LP}(G - Y)$, where $Y$ is a minimal blocking set whose size we want to bound, and by considering the complete graph with $d+2$ and $2d+2$ vertices, we showed that $\mathcal{C}_{d\text{-qf}} = d+2$, that $\mathcal{C}_{d\text{-qb}} = d+2$, and that $\mathcal{C}_{d\text{-qi}} = 2d+2$.

Afterwards, we generalized this result for robust hereditary graph classes $\mathcal{C}$ where $\beta_{\mathcal{C}}$ is bounded, by showing that for all $d \geq 1$ it holds that $\beta_{\mathcal{C}}(d) = 2^{d-1} + 1$, if $\beta_{\mathcal{C}} = 1$ and that $\beta_{\mathcal{C}}(d) = (\beta_{\mathcal{C}} - 1)2^d + 1$, if $\beta_{\mathcal{C}} \geq 2$. We were also able to obtain a bound for the value $\beta_{\mathcal{C}}(d)$ for some non-hereditary graph classes if these classes are $f$-solid. More precisely, we showed that $\beta_{\mathcal{C}}(d) \leq \left(\sum_{i=0}^{d} \binom{d}{i} f(i)\right) - 2^d + 1$, if $\mathcal{C}$ is a robust, $f$-solid graph class where $\beta_{\mathcal{C}}$ is bounded. This bound is weaker than the bound for robust, hereditary graph classes and we do not know whether this bound is tight.

[5]The graph class $\mathcal{C}_{\text{LP}}$ is robust because for every graph $G$ with $\text{VC}(G) = \text{LP}(G)$ it holds for every connected component $H$ of $G$ that $\text{VC}(H) = \text{LP}(H)$.

# CHAPTER 6 KERNELIZATION FOR VERTEX COVER

## 6.1. Introduction

In this chapter, we will combine the results from Chapter 4 and Chapter 5 to obtain polynomial kernelizations for VERTEX COVER parameterized by the size of a $\mathcal{C}$-modulator or a $(\mathcal{C}, d)$-modulator.

First, we use the bounds for the minimal blocking set size of Section 5.2 to obtain polynomial kernels for VERTEX COVER parameterized by the size of $\mathcal{C}$-modulator where $\mathcal{C}$ is the class of $d$-quasi-forests, $d$-quasi-bipartite graphs or $d$-quasi-integral graphs (Section 6.2). This generalizes, for example, the polynomial kernel for VERTEX COVER parameterized by the size of a modulator to a pseudoforest [FS16]. Furthermore, using Theorem 4.3 we even improve the kernel size for VERTEX COVER parameterized by the size of a modulator to a pseudoforest.

Second, our bounds for the minimal blocking set size relative to elimination distances (Section 5.3) allow us to generalize and combine previous polynomial kernelization results for VERTEX COVER (Section 6.3). We state this explicitly for elimination distances to hereditary graph classes.

**Theorem 6.1.** *Let $\mathcal{C}$ be a hereditary and robust graph class for which $\beta_{\mathcal{C}}$ is bounded, such that* VERTEX COVER *has a (randomized) polynomial kernelization parameterized by the size of a modulator to $\mathcal{C}$. Then* VERTEX COVER *also has a (randomized) polynomial kernelization parameterized by the size of a modulator to graphs of bounded elimination distance to $\mathcal{C}$.*

As an example, this combines known polynomial kernels relative to the size of modulators to a forest [JB13] respectively to graphs of bounded treedepth [BS19] to polynomial kernels relative to a modulator to graphs of bounded forest elimination distance. Similarly, the randomized polynomial kernel for VERTEX COVER parameterized by a modulator to bipartite graphs is generalized to a modulator to graphs of bounded bipartite elimination distance. The approach to this result (also in the non-hereditary case) uses our bounds for minimal blocking set size relative to elimination distances and, apart from that, is inspired by the result of Bougeret and Sau [BS19]. Intuitively, these kernels are obtained by suitable reductions to the known kernelizable cases, and thus carry over their properties (e.g., being deterministic or randomized).

As an explicit example for the non-hereditary case, we state a new kernelization result relative to the size of a modulator to the class of graphs of bounded elimination distance to $\mathcal{C}_{\text{LP}}$, i.e., bounded elimination distance to graphs where optimum vertex cover size equals optimum fractional vertex cover size (Section 6.3.2).

**Theorem 6.2.** Vertex Cover *admits a randomized polynomial kernel parameterized by the size of a modulator to graphs that have bounded elimination distance to $\mathcal{C}_{\text{LP}}$.*

This result subsumes several polynomial kernelizations for Vertex Cover (except for their size bounds).

## 6.2. $\mathcal{C}$-Modulators

To obtain kernelizations for Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator, $\mathcal{C}_{d\text{-qb}}$-modulator, or $\mathcal{C}_{d\text{-qi}}$-modulator we use the kernelizations for Vertex Cover parameterized by the size of a feedback vertex set [JB13], the size of an odd cycle transversal [KW12], or the difference between an optimum vertex cover and an optimum LP solution [KW12], respectively.

**Theorem 6.3.** *Let $\mathcal{C}_{d\text{-qf}}$ be the class of $d$-quasi-forests.* Vertex Cover *parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator $X$ admits a kernel with $\mathcal{O}(|X|^3 + d^3|X|^{3d+6})$ vertices.*

**Proof.** Let $(G, k, X)$ be an instance of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator. It follows from Theorem 4.3 that there exists a polynomial time algorithm that given instance $(G, k, X)$ of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator returns an equivalent instance $(\widetilde{G}, \widetilde{k}, X)$ of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator where $\widetilde{G} - X$ has at most $|X|^{\beta_{\mathcal{C}_{d\text{-qf}}}} = |X|^{d+2}$ connected components, because $\mathcal{C}_{d\text{-qf}}$ is a hereditary graph class with $\beta_{\mathcal{C}_{d\text{-qf}}} = d + 2$ on which vertex cover is solvable in polynomial time.

Now, we add for each of the at most $|X|^{d+2}$ connected components of $\widetilde{G} - X$ a feedback vertex set of size at most $d$ to $X$. Let $Z$ be the union of these feedback vertex sets. It holds that $Z$ contains at most $d \cdot |X|^{d+2}$ vertices, namely, at most $d$ vertices from each of the $|X|^{d+2}$ connected components of $\widetilde{G} - X$. We add the vertex set $Z$ to the modulator $X$ to obtain an instance $(\widetilde{G}, \widetilde{k}, \widetilde{X} = X \cup Z)$ of Vertex Cover parameterized by the size of a $\mathcal{C}_{forest}$-modulator, where $\mathcal{C}_{forest}$ is the class of forests.

The instances $(\widetilde{G}, \widetilde{k}, X)$ and $(\widetilde{G}, \widetilde{k}, \widetilde{X})$ are obviously equivalent. To prove that we can construct $(\widetilde{G}, \widetilde{k}, \widetilde{X})$ in polynomial time, we only have to show that we can find the set

$Z$ in polynomial time. This holds, because we can find a feedback vertex of constant size in polynomial time.

Finally, we apply the kernelization algorithm for Vertex Cover parameterized by the size of a $\mathcal{C}_{forest}$-modulator of Jansen and Bodlaender [JB13] to the instance $(\widetilde{G}, \widetilde{k}, \widetilde{X})$ of Vertex Cover parameterized by the size of a $\mathcal{C}_{forest}$-modulator to obtain in polynomial time an equivalent instance $(G', k', X')$ of Vertex Cover parameterized by the size of a $\mathcal{C}_{forest}$-modulator. Obviously, $(G', k', X')$ is also an instance of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator.

To conclude the proof, we have to bound the number of vertices in $G'$. It holds that instance $(G', k', X')$ has at most $2|\widetilde{X}| + 28|\widetilde{X}|^2 + 56|\widetilde{X}|^3$ vertices [JB13, Corollary 2]. Since $\widetilde{X} = X \cup Z$, and since $|Z| \leq d \cdot |X|^{d+2}$, this leads to the desired bound for the size of $V(G')$:

$$\begin{aligned} |V(G')| &\leq 2|\widetilde{X}| + 28|\widetilde{X}|^2 + 56|\widetilde{X}|^3 \\ &\leq 2(d \cdot |X|^{d+2} + |X|) + 28(d \cdot |X|^{d+2} + |X|)^2 + 56(d \cdot |X|^{d+2} + |X|)^3 \\ &\in \mathcal{O}(|X|^3 + d^3 |X|^{3d+6}). \end{aligned}$$

This concludes the proof. ■

The kernelization for Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qb}}$-modulator is similar to the kernelization for Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator. The only difference is that we apply the kernelization for Vertex Cover parameterized by an odd cycle transversal instead of parameterized by a feedback vertex set after reducing the number of connected components.

**Theorem 6.4.** *Let $\mathcal{C}_{d\text{-}qb}$ be the class of $d$-quasi-bipartite graphs.* Vertex Cover *parameterized by the size of a $\mathcal{C}_{d\text{-}qb}$-modulator $X$ admits a randomized polynomial kernel with $\mathcal{O}(|X|^6 + d^6 \cdot |X|^{6d+12})$ vertices.*

**Proof.** Let $(G, k, X)$ be an instance of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qb}}$-modulator. As for the kernelization for Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator, we can apply Theorem 4.3 to obtain in polynomial time an equivalent instance $(\widetilde{G}, \widetilde{k}, X)$ of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qb}}$-modulator where $\widetilde{G} - X$ has at most $|X|^{\beta_{\mathcal{C}_{d\text{-qf}}}} = |X|^{d+2}$ connected components, because $\mathcal{C}_{d\text{-qb}}$ is a hereditary graph class on which Vertex Cover is solvable in polynomial time, and because $\beta_{\mathcal{C}_{d\text{-qb}}}$ is bounded.

Next, we add for each connected component of $\widetilde{G} - X$ an odd cycle transversal of size at most $d$ to the modulator $X$. Let $Z$ be the union of these odd cycle transversals. Observe that $|Z| \leq d \cdot |X|^{d+2}$ (as before) and that $\widetilde{G} - (X \cup Z)$ is a bipartite graph.

Finally, we apply the randomized polynomial kernelization algorithm for Vertex Cover parameterized by the size of a modulator to a bipartite graph due to Kratsch and Wahlström [KW12] to the instance $(\widetilde{G}, \widetilde{k}, \widetilde{X} = X \cup Z)$. This leads to an instance $(G', k', X')$ of Vertex Cover parameterized by the size of a modulator to a bipartite graph where $V(G')$ is polynomially bounded in $|\widetilde{X}|$. Since $\widetilde{X} = X \cup Z$ it follows that $|\widetilde{X}| \leq |X| + d \cdot |X|^{d+2}$. Overall, this leads to the desired bound for $|V(G')| = \mathcal{O}(|\widetilde{X}|^6) = \mathcal{O}(|X|^6 + d^6 \cdot |X|^{6d+12})$ and concludes the proof. ■

The kernelization for Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qi}}$-modulator differs a little bit from the previous ones. After reducing the number of instances we have to show that we can bound the difference between an optimum vertex cover and an optimum LP solution in the remaining graph using the size of the $\mathcal{C}_{d\text{-qi}}$-modulator $X$. It does not help to add some vertices to the modulator because we want to apply the kernel for Vertex Cover parameterized by the difference between an optimum vertex cover and an optimum LP solution.

**Theorem 6.5.** *Let $\mathcal{C}_{d\text{-}qi}$ be the class of d-quasi-integral graphs.* Vertex Cover *parameterized by the size of a $\mathcal{C}_{d\text{-}qi}$-modulator admits a randomized polynomial kernel with $\mathcal{O}(|X|^6 + d^6 \cdot |X|^{12d+12})$ vertices.*

**Proof.** Let $(G, k, X)$ be an instance of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qi}}$-modulator. The graph class $\mathcal{C}_{d\text{-qi}}$ is not hereditary but robust by definition. We will show that we can solve Vertex Cover in polynomial time on graphs of graph class $\mathcal{C}_{d\text{-qi}}+1$, by proving that every graph of graph class $\mathcal{C}_{d\text{-qi}}+1$ is $(d+1)$-quasi-integral. Let $G$ be a graph of graph class $\mathcal{C}_{d\text{-qi}} + 1$, and let $H$ be an arbitrary connected component of $G$. Since $G$ is contained in graph class $\mathcal{C}_{d\text{-qi}} + 1$ there exists a vertex $v_H \in V(H)$ such that $H - v_H \in \mathcal{C}_{d\text{-qi}}$. This implies that $\textsc{vc}(H - v_H) \leq \textsc{lp}(H - v_H) + d$. Obviously, it holds that $\textsc{vc}(H) \leq \textsc{vc}(H - v_H) + 1$, and that $\textsc{lp}(H - v_H) \leq \textsc{lp}(H)$. Combining these three inequalities we obtain that $\textsc{vc}(H) \leq \textsc{vc}(H - v_H) + 1 \leq \textsc{lp}(H - v_H) + d + 1 \leq \textsc{lp}(H) + d + 1$. Thus, for every connected component $H$ of $G$ it holds that $\textsc{vc}(H) \leq \textsc{lp}(H) + d + 1$ which implies that $G$ is $(d + 1)$-quasi-integral.

Since $d$ is a constant and Vertex Cover is fixed-parameter tractable when parameterized by the difference between an optimum vertex cover and an optimum LP solution, we can compute a vertex cover of every graph in graph class $\mathcal{C}_{d\text{-qi}}$, as well as $\mathcal{C}_{d\text{-qi}}+1$, in polynomial time. Thus, we can apply Theorem 4.12 to obtain in polynomial time an equivalent instance $(\widetilde{G}, \widetilde{k}, X)$ of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qi}}$-modulator, where $\widetilde{G} - X$ has at most $|X|^{2d+2}$ connected components.

Note that a vertex cover in $\widetilde{G} - X$ together with the set $X$ is a vertex cover of $\widetilde{G}$ and that the size of this vertex cover is $\textsc{vc}(\widetilde{G} - X) + |X|$. We can assume that $\widetilde{k}$ is strictly smaller than the size of this vertex cover. Otherwise the set $X \cup \widetilde{S}$ is a vertex cover of $G$ of size at most $\widetilde{k}$ that we can compute in polynomial time, where $\widetilde{S}$ is a minimum vertex cover of $\widetilde{G} - X$. Thus, in the following we assume that $\textsc{vc}(\widetilde{G} - X) + |X| > \widetilde{k}$.

Finally, we apply the randomized kernelization algorithm for Vertex Cover above lp [KW12] to the instance $(\widetilde{G}, \widetilde{k})$ to obtain in polynomial time an instance $(G', k')$ of Vertex Cover above lp where $V(G')$ has size at most $\mathcal{O}((\widetilde{k} - \textsc{lp}(\widetilde{G}))^6)$. Next, we will show that we can bound $\widetilde{k} - \text{LP}(\widetilde{G})$ polynomially in the size of $X$:

$$\begin{aligned}
\widetilde{k} - \text{LP}(\widetilde{G}) &\le \widetilde{k} - \text{LP}(\widetilde{G} - X) \\
&= \widetilde{k} - \sum_{H \text{ c.c of } \widetilde{G}-X} \text{LP}(H) \\
&\le \widetilde{k} - \sum_{H \text{ c.c of } \widetilde{G}-X} (\text{VC}(H) - d) \\
&< \text{VC}(\widetilde{G} - X) + |X| - \sum_{H \text{ c.c of } \widetilde{G}-X} \text{VC}(H) + \sum_{H \text{ c.c of } \widetilde{G}-X} d \\
&= \text{VC}(\widetilde{G} - X) + |X| - \text{VC}(\widetilde{G} - X) + |X|^{2d+2} d \\
&\le |X| + |X|^{2d+2} d
\end{aligned}$$

This implies that $V(G')$ has at most $\mathcal{O}((|X| + |X|^{2d+2}d)^6) = \mathcal{O}(|X|^6 + d^6 \cdot |X|^{12d+12})$ vertices. Now, the instance $(G', X' = V(G'), k')$ is an equivalent instance of Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qi}}$-modulator that has the desired number of vertices. This concludes the proof. ■

## 6.3. $(\mathcal{C}, d)$-Modulators

In Section 4.4.1 we have seen necessary assumptions on a graph class $\mathcal{C}$ such that the number of connected components outside the $\mathcal{C}$-modulator could be efficiently reduced. We want to apply Reduction Rule 4.1 also for instances of Vertex Cover parameterized by the size of a $(\mathcal{C}, d)$-modulator. Hence, we start by extending these results to using $(\mathcal{C}, d)$-modulators, instead of $\mathcal{C}$-modulators.

**Lemma 6.6.** *Let $\mathcal{C}$ be a hereditary graph class on which* Vertex Cover *is polynomial-time solvable. In polynomial time we can compute an optimum* Vertex Cover *of a graph $G$ with $\text{ed}_{\mathcal{C}}(G) \le d$, when $d$ is a fixed constant.*

**Proof.** We prove this via induction over the elimination distance of graph $G$. Obviously, if $\text{ed}_{\mathcal{C}}(G) = 0$ then $G$ is a graph of the class $\mathcal{C}$; thus we can compute an optimum vertex cover in polynomial time.

For the induction step, we assume that we can solve Vertex Cover in polynomial time on graphs $G$ with $\text{ed}_{\mathcal{C}}(G) < d$.

Assume that $d = \text{ed}_{\mathcal{C}}(G) > 0$. Let $G_1, G_2, \ldots, G_h$ be the connected components of $G$. It is enough to compute the optimum vertex cover of each connected component $G_i$ of $G$ with $1 \le i \le h$ because $\text{VC}(G) = \sum_{i=1}^{h} \text{VC}(G_i)$. Let $r_i$ be the root of an optimum elimination tree $F_i$ of graph $G_i$. We distinguish between the case that $r_i$ is contained in the optimum vertex cover or not. Hence,

$$\text{VC}(G_i) = \min\{\text{VC}(G_i - r_i) + 1, \text{VC}(G_i - N[r_i]) + |N(r_i)|\}.$$

Observe that the graphs $G_i - r_i$ and $G_i - N[r_i]$ have elimination distance less than $d$ to $\mathcal{C}$: This is clear for $G_i - r_i$ by removing $r_i$ from the elimination tree $F_i$ of $G_i$. For $G_i - N[r_i]$ we can similarly remove $N[r_i] \ni r_i$ from the elimination tree $F_i$ of $G_i$ to see this, using that $\mathcal{C}$ is hereditary. By the inductive assumption we can compute an optimum vertex cover for both graphs in polynomial time, which shows that we can compute an optimum vertex cover of $G$ in polynomial time. Note that while the running time is polynomial for $d$ constant, it may depend exponentially on $d$. ∎

**Corollary 6.7.** *Reduction Rule 4.1 is applicable in polynomial time on graphs $G$ with a given $(\mathcal{C}, d)$-modulator $X$, where $\mathcal{C}$ is a hereditary graph class on which* VERTEX COVER *is solvable in polynomial time, and where $\beta_{\mathcal{C}}$ is bounded.*

**Proof.** Since $\mathcal{C}$ is hereditary and VERTEX COVER is solvable in polynomial time on $\mathcal{C}$, it follows from Lemma 6.6 that we can efficiently solve VERTEX COVER in graphs $G$ with $\mathrm{ed}_{\mathcal{C}}(G) \leq d$. Furthermore, it follows from the fact that $\beta_{\mathcal{C}}$ is bounded and Theorem 5.25 that $\beta_{\mathcal{C}}(d)$ is bounded. Finally, it follows from Lemma 4.10 that we can apply Reduction Rule 4.1 in polynomial time, because the class of graphs with elimination distance at most $d$ to a hereditary graph class $\mathcal{C}$ is also hereditary. ∎

For non-hereditary graph classes $\mathcal{C}$ we again need that VERTEX COVER is solvable in polynomial time on the graph class $\mathcal{C} + c$ with $c$ constant. Here $\mathcal{C} + 1$ is not enough because in each recursive step we add a vertex to some of the base components.

**Lemma 6.8.** *Let $\mathcal{C}$ be a robust graph class on which* VERTEX COVER *is polynomial-time solvable. Furthermore, assume that* VERTEX COVER *is polynomial-time solvable on the graph class $\mathcal{C} + c$ for every constant $c \in \mathbb{N}$. In polynomial time we can compute an optimum* VERTEX COVER *of a graph $G$ with $\mathrm{ed}_{\mathcal{C}}(G) \leq d$.*

**Proof.** Again, we use induction to prove the lemma. The construction is similar to the construction of Lemma 5.26. For the base case assume that $\mathrm{ed}_{\mathcal{C}}(G) = 0$. This implies that $G$ is a graph of the class $\mathcal{C}$. Hence, we can compute an optimum vertex cover in polynomial time.

For the induction step, we assume that we can solve VERTEX COVER in polynomial time on graphs $G$ with $\mathrm{ed}_{\mathcal{C}}(G) < d$ for all graph classes $\mathcal{C}$ that fulfill the requirements of the lemma.

Let $G$ be a graph with $\mathrm{ed}_{\mathcal{C}}(G) = d > 0$. Let $G_1, G_2, \ldots, G_h$ be the connected components of $G$. Again, it is sufficient to compute the optimum vertex cover of each connected component separately. Let $G_i$ be a connected component of $G$ and let $r_i$ be the root of an optimum elimination tree $F_i$ of $G_i$. As before, we want to compute an optimum vertex cover of $G_i - r_i$ and $G_i - N[r_i]$. Since $r_i$ is the root of the elimination tree of $G_i$ it holds that $\mathrm{ed}_{\mathcal{C}}(G_i - r_i) = d - 1$. Hence, we can compute an optimum vertex cover of $G_i - r_i$ in polynomial time for $d$ constant.

To compute an optimum vertex cover of $G_i - N[r_i]$ we construct the graph $\hat{G}_i$ as in the proof of Lemma 5.26. Let $Z' \subseteq N(r_i)$ the set of vertices in $N(r_i)$ that are contained in the base components of the elimination tree $F_i$ of $G_i$. It holds that $\mathrm{ed}_{\mathcal{C}+1}(\hat{G}_i) \leq d-1$. Observe that $\mathrm{VC}(\hat{G}_i) \leq \mathrm{VC}(G_i - N[r_i]) + |Z'|$ because every optimum vertex cover of $G_i - N[r_i]$ together with the set $Z'$ is a vertex cover of $\hat{G}_i$. If $N(r_i)$ is not a blocking set of $G_i$ then $\mathrm{VC}(G_i - N[r_i]) = \mathrm{VC}(\hat{G}_i) - |Z'|$ (see proof of Lemma 5.26).

If $N(r_i)$ is a blocking set of $G_i$ then there exists an optimum vertex cover of $G_i$ that contains $r_i$. Furthermore, it holds that $\mathrm{VC}(\hat{G}_i) - |Z'| + |N(r_i)| \geq \mathrm{VC}(G_i)$ when $N(r_i)$ is a blocking set of $G_i$: Let $\hat{S}$ be an optimum vertex cover of $\hat{G}_i$. We add all vertices in $N(r_i) \setminus Z'$ to $\hat{S}$ and denote the resulting set by $S'$. Recall that $Z' \subseteq N(r_i)$ which implies that $|S'| = |\hat{S}| + |N(r_i)| - |Z'|$. If $Z' \subseteq \hat{S}$ then $S'$ is a vertex cover of $G_i$ that contains $N(r_i)$. This implies that $|S'| > \mathrm{VC}(G_i)$ because $N(r_i)$ is a blocking set of $G_i$. Since $|S'| = |\hat{S}| + |N(r_i) \setminus Z'|$ we obtain that $\mathrm{VC}(G_i) < \mathrm{VC}(\hat{G}) - |Z'| + |N(r_i)|$. If $Z' \nsubseteq \hat{S}$ then $\hat{S}$ contains some newly added vertices. Hence, $S = S' \cap V(G_i) \cup \{r\}$ is a vertex cover of $G_i$ of size at most $|S'|$. It holds that $\mathrm{VC}(G_i) \leq |S| = |S' \cap V(G_i)| + |\{r\}| \leq |S'| = |\hat{S}| + |N(r_i) \setminus Z'| = \mathrm{VC}(\hat{G}) - |Z'| + |N(r_i)|$.

Overall, this implies that $\mathrm{VC}(G_i) = \min\{\mathrm{VC}(G_i - N[r_i]) + |N(r_i)|, \mathrm{VC}(G_i - r_i) + 1\} = \min\{\mathrm{VC}(\hat{G}_i) - |Z'| + |N(r_i)|, \mathrm{VC}(G_i - r_i) + 1\}$. As mentioned above, it holds that $\mathrm{ed}_{\mathcal{C}+1}(\hat{G}_i) \leq d - 1$. The graph class $\mathcal{C} + 1$ fulfills the desired properties of the lemma if $\mathcal{C}$ fulfills these properties, because $(\mathcal{C} + 1) + 1 = \mathcal{C} + 2$. Thus, we can compute an optimum vertex cover of $\hat{G}_i$ in polynomial time for constant $d$. Again, the running time may depend exponentially on $d$. ■

For the next corollary, observe that in particular $\beta_{\mathcal{C}}(d)$ is bounded if $\mathcal{C}$ is known to be either hereditary or $f$-solid, by Theorems 5.17 and 5.25.

**Corollary 6.9.** *Reduction Rule 4.1 is applicable in polynomial time on graphs $G$ with a given $(\mathcal{C}, d)$-modulator $X$, where $\mathcal{C}$ is a robust graph class with the properties that $\beta_{\mathcal{C}}(d)$ is bounded for any constant $d$ and that* Vertex Cover *is polynomial-time solvable on the graph class $\mathcal{C} + c$ for constant $c$.*

**Proof.** It follows from Lemma 6.8 that we can solve Vertex Cover in polynomial time on graphs $G$ with $\mathrm{ed}_{\mathcal{C}}(G) \leq d$ for any constant $d$. It then follows from Lemma 4.11 that we can verify whether a given set of vertices is a blocking set in $G$ with $\mathrm{ed}_{\mathcal{C}}(G) \leq d$, as adding a single vertex to $G$ increases its elimination distance to $\mathcal{C}$ by at most one. The statement now follows from Lemma 4.9. ■

### 6.3.1. General Results

In this section, we show that Vertex Cover parameterized by the size of a $(\mathcal{C}, d)$-modulator has a polynomial kernel when the graph class $\mathcal{C}$ fulfills some additional properties. The assumptions that $\beta_{\mathcal{C}}(d)$ is bounded and that Vertex Cover is polynomial-time solvable on the considered graph class are necessary, if these assumptions fail a polynomial kernel is unlikely to exist. The same holds for the assumption that Vertex Cover parameterized by a $\mathcal{C}$-modulator has a polynomial kernel. We additionally require that $\mathcal{C}$ is a robust graph class that is either hereditary, or has the property that

Vertex Cover is polynomial-time solvable on $\mathcal{C}+c$. These assumptions will ensure that our reduction rule can be applied in polynomial time.

**Lemma 6.10.** *Let $\mathcal{C}$ be a robust graph class for which $\beta_{\mathcal{C}}(d)$ is bounded and on which* Vertex Cover *is polynomial-time solvable, such that $\mathcal{C}$ is hereditary or* Vertex Cover *is polynomial-time solvable on $\mathcal{C}+c$ for all constants $c$.*

*Suppose* Vertex Cover *parameterized by the size of a $\mathcal{C}$-modulator $\hat{X}$ has a (randomized) polynomial kernel with $g(|\hat{X}|)$ vertices. Then* Vertex Cover *parameterized by the size of a $(\mathcal{C},d)$-modulator $X$ has a (randomized) polynomial kernel with $\mathcal{O}(g(|X|^b))$ vertices, where $b=\prod_{i=1}^{d}\beta_{\mathcal{C}}(i)$.*

Our kernelization for Vertex Cover parameterized by the size of a $(\mathcal{C},d)$-modulator is similar to the kernelization for Vertex Cover parameterized by the size of a $d$-treedepth modulator (see [BS19]). One difference is that we do not want to introduce hyper-edges. For completeness, we give a short proof of Lemma 6.10.

**Proof of Lemma 6.10.** Like Bougeret and Sau [BS19] we reduce an instance $(G,k,X)$ of Vertex Cover parameterized by the size of a $(\mathcal{C},d)$-modulator to an instance $(G',k',X')$ of Vertex Cover parameterized by the size of a $(\mathcal{C},d-1)$-modulator. The bound on the number of vertices follows inductively.

We start by observing that Reduction Rule 4.1 can be applied in polynomial time: If $\mathcal{C}$ is hereditary, it follows from Corollary 6.7 that Reduction Rule 4.1 can be applied in polynomial time. Otherwise, Vertex Cover is polynomial-time solvable on graphs from $\mathcal{C}+c$ for any constant $c$ and it follows immediately from Corollary 6.9 that Reduction Rule 4.1 can be applied in polynomial time.

To obtain the kernel, we first apply Reduction Rule 4.1 to instance $(G,k,X)$ of Vertex Cover parameterized by the size of a $(\mathcal{C},d)$-modulator. This leads to an equivalent instance $(G',k',X)$ of Vertex Cover parameterized by the size of a $(\mathcal{C},d)$-modulator where the number of connected components in $G'-X$ is at most $|X|^{\beta_{\mathcal{C}}(d)}$ (Theorem 4.8). Let $X_r$ be the set of roots of an optimum elimination forest $F$ of $G'-X$. Thus, $X_r$ contains at most $|X|^{\beta_{\mathcal{C}}(d)}$ vertices because every connected component of $G'-X$ has exactly one root. Now, $(G',k',X\cup X_r)$ is an instance of Vertex Cover parameterized by the size of a $(\mathcal{C},d-1)$-modulator. Obviously, $(G,k,X)$ is a yes-instance if and only if $(G',k',X\cup X_r)$ is a yes-instance.

It follows inductively that we can reduce the instance $(G',k',X\cup X_r)$ of Vertex Cover parameterized by the size of a $(\mathcal{C},d-1)$-modulator to an instance $(\hat{G},\hat{k},\hat{X})$ of Vertex Cover parameterized by the size of a $(\mathcal{C},0)$-modulator with

$$|\hat{X}|\in\mathcal{O}\left(|X\cup X_r|^{\prod_{i=1}^{d-1}\beta_{\mathcal{C}}(i)}\right)=\mathcal{O}\left(\left(|X|^{\beta_{\mathcal{C}}(d)}\right)^{\prod_{i=1}^{d-1}\beta_{\mathcal{C}}(i)}\right)=\mathcal{O}\left(|X|^{\prod_{i=1}^{d}\beta_{\mathcal{C}}(i)}\right).$$

Since $(\hat{G},\hat{k},\hat{X})$ is also an instance of Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator, we can reduce instance $(\hat{G},\hat{k},\hat{X})$ to an equivalent instance $(\widetilde{G},\widetilde{k},\widetilde{X})$ of

Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator with $\mathcal{O}(g(|X|^b))$ vertices where $b = \prod_{i=1}^{d} \beta_{\mathcal{C}}(i)$. This concludes the proof. ■

Observe that in the above lemma statement, when Vertex Cover parameterized by a $\mathcal{C}$-modulator has a polynomial kernel, the fact that Vertex Cover is solvable in polynomial time on graphs from $\mathcal{C} + c$ is immediate because the polynomial kernel implies that the problem is fixed-parameter tractable in the parameter. Since in this case the size of a $\mathcal{C}$-modulator of each connected component is $c$, which is constant, the result follows.

In the above theorem statement, we assume that $\beta_{\mathcal{C}}(d)$ is bounded to obtain the kernelization. We observe that for hereditary graph classes, this assumption is not needed, as it follows from our results in Theorem 5.17 that it suffices to bound $\beta_{\mathcal{C}}$. Furthermore, a bound on $\beta_{\mathcal{C}}$ often comes naturally: if Vertex Cover parameterized by a $\mathcal{C}$-modulator has a polynomial kernel, it follows from Theorem 4.1 that, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, there must exist a constant $d$ such that $\beta_{\mathcal{C}} \leq d$.

**Theorem 6.1.** *Let $\mathcal{C}$ be a hereditary and robust graph class for which $\beta_{\mathcal{C}}$ is bounded, such that* Vertex Cover *has a (randomized) polynomial kernelization parameterized by the size of a modulator to $\mathcal{C}$. Then* Vertex Cover *also has a (randomized) polynomial kernelization parameterized by the size of a modulator to graphs of bounded elimination distance to $\mathcal{C}$.*

**Proof.** The result is immediate from Lemma 6.10, combined with the bound on $\beta_{\mathcal{C}}(d)$ for hereditary graph classes provided in Theorem 5.17. ■

Similarly, for non-hereditary graph classes, it suffices if $\mathcal{C}$ is $f$-solid to obtain a polynomial kernel. The size of the kernel depends on $f$.

**Theorem 6.11.** *Let $\mathcal{C}$ be a robust and $f$-solid graph class for which $\beta_{\mathcal{C}}$ is bounded and for which* Vertex Cover *parameterized by the size of a $\mathcal{C}$-modulator has a (randomized) polynomial kernel. Then* Vertex Cover *parameterized by the size of a $(\mathcal{C}, d)$-modulator has a (randomized) polynomial kernel.*

**Proof.** The result is immediate from Lemma 6.10, combined with the bound on $\beta_{\mathcal{C}}(d)$ for $f$-solid graph classes provided in Theorem 5.25. ■

### 6.3.2. Kernel for Modulator to Bounded $\mathcal{C}_{\mathrm{LP}}$ Elimination Distance

In this section, we show how Theorem 6.2 follows from the general results in the previous section, to have an explicit example for a non-hereditary base class $\mathcal{C}$. That is, we show how to get a randomized polynomial kernel for Vertex Cover parameterized by the size of a modulator $X$ such that $G - X$ has bounded elimination distance to the non-hereditary class $\mathcal{C}_{\mathrm{LP}}$ of graphs where integral and fractional vertex cover number coincide.

Combining Corollary 5.27 and Lemma 6.10 we can now generalize the kernelization for Vertex Cover parameterized by the size of a $d$-treedepth modulator and parameterized by the difference between an optimum vertex cover and an optimum LP solution using the size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator as the parameter. The following theorem subsumes Theorem 6.2.

**Theorem 6.12.** *An optimum $(\mathcal{C}_{\text{LP}}, d)$-modulator of a graph $G$ has at most the size of a $d$-treedepth modulator of $G$ and at most twice the size of $\text{VC}(G) - \text{LP}(G)$. Furthermore,* Vertex Cover *parameterized by the size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator admits a randomized polynomial kernel.*

**Proof.** The empty graph is contained in $\mathcal{C}_{\text{LP}}$ because both integral and fractional vertex cover number are zero. It follows directly that the elimination distance to $\mathcal{C}_{\text{LP}}$ is upper bounded by the treedepth, i.e., $\text{ed}_{\mathcal{C}_{\text{LP}}}(G) \leq \text{td}(G)$, and that every $d$-treedepth modulator of $G$ is also a $(\mathcal{C}_{\text{LP}}, d)$-modulator of $G$.

We showed in Lemma 3.8 that there exists a $\mathcal{C}_{\text{LP}}$-modulator in $G$ of size at most $2 \cdot (\text{VC}(G) - \text{LP}(G))$. This modulator is also a $(\mathcal{C}_{\text{LP}}, 0)$-modulator of $G$. Hence, the size of an optimum $(\mathcal{C}_{\text{LP}}, d)$-modulator of $G$ is at most the size of a $\mathcal{C}_{\text{LP}}$-modulator of $G$ which is at most $2 \cdot (\text{VC}(G) - \text{LP}(G))$.

Now, we will show that Vertex Cover parameterized by the size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator admits a randomized polynomial kernel. It holds that $\beta_{\mathcal{C}}(d)$ is bounded (see Corollary 5.27). Furthermore, Vertex Cover parameterized by the size of a $\mathcal{C}_{\text{LP}}$-modulator admits a randomized polynomial kernel because the size of the modulator is at most the difference between an optimum vertex cover and an optimum LP solution, and because Vertex Cover parameterized by the difference between an optimum vertex cover and an optimum LP solution has a randomized polynomial kernel [KW12].

Furthermore, we can show that Vertex Cover is solvable in polynomial time on graphs from $\mathcal{C}_{\text{LP}} + c$. Let $G$ be a graph from $\mathcal{C}_{\text{LP}} + c$. Let $H$ be any connected component of $G$, and let $X_H \subseteq V(H)$ with $|X_H| \leq c$ such that $H - X_H$ is connected and contained in graph class $\mathcal{C}_{\text{LP}}$. It holds that $\text{VC}(H) \leq \text{VC}(H - X_H) + |X_H|$ and that $\text{LP}(H - X) \leq \text{LP}(H)$. Thereby, $\text{VC}(H) - \text{LP}(H) \leq \text{VC}(H - X_H) + |X_H| - \text{LP}(H - X_H) \leq c$ because $H - X_H \in \mathcal{C}_{\text{LP}}$ which implies that $\text{VC}(H - X_H) = \text{LP}(H - X_H)$. Since Vertex Cover parameterized by $\text{VC}(H) - \text{LP}(H)$ is fixed-parameter tractable [NRRS12], it follows that when $\text{VC}(H) - \text{LP}(H)$ is constant, the problem is solvable in polynomial time. Overall, we showed that Vertex Cover is solvable in polynomial time on each connected component of $G \in \mathcal{C}_{\text{LP}} + c$ which implies that Vertex Cover is solvable in polynomial time on graphs from $\mathcal{C}_{\text{LP}} + c$.

Thus, it follows from Lemma 6.10 that Vertex Cover parameterized by the size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator admits a randomized polynomial kernel. ■

As mentioned above, Theorem 6.12 shows that choosing as parameter the size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator, generalizes two very general parameters for Vertex Cover that are incomparable, namely the parameters size of a $d$-treedepth-modulator (with $d$ constant) and the difference between an optimum vertex cover and an optimum LP solution. This indicates that the size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator is a very general parameter for Vertex Cover. Furthermore, it generalizes all, except one, kernelization results for Vertex Cover. The only parameter that is not generalized by this parameter, and for which a polynomial kernel is known, is the parameter $\ell = 2\text{LP} - \text{MM}$.

## 6.4. Summary

In this chapter we combined the results of Chapter 4 and Chapter 5 to obtain polynomial kernels for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator or a $(\mathcal{C}, d)$-modulator. First of all, we generalized and improved the result of Fomin and Strømme [FS16] by showing that Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qf}}$-modulator has a kernel with $\mathcal{O}(d^3 \cdot |X|^{3d+6})$ vertices. Next, we showed that Vertex Cover parameterized by the size of a $\mathcal{C}_{d\text{-qb}}$-modulator or the size of a $\mathcal{C}_{d\text{-qi}}$-modulator has a randomized polynomial kernel.

Let $\mathcal{C}$ be a robust graph class where $\beta_{\mathcal{C}}$ is bounded and where $\mathcal{C}$ is either hereditary or $f$-solid. We showed that Vertex Cover parameterized by the size of a $(\mathcal{C}, d)$-modulator has a (randomized) polynomial kernel if Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator has a (randomized) polynomial kernel. This implies, for example, that Vertex Cover parameterized by the size of a $(\mathcal{C}_{forest}, d)$-modulator $X$ has a polynomial kernel with $\mathcal{O}(|X|^{3b})$ vertices where $b = \prod_{i=1}^{d}(2^i + 1)$ and that Vertex Cover parameterized by the size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator has a randomized polynomial kernel. Furthermore, we showed that the parameter size of a $(\mathcal{C}_{\text{LP}}, d)$-modulator generalizes the two incomparable parameters size of a $d$-treedepth modulator as well as size of a $\mathcal{C}_{\text{LP}}$-modulator.

# CHAPTER 7 CONCLUSION AND OPEN PROBLEMS

In this chapter we recapitulate the results of this part and state some open problems. Motivated by the large number of positive and negative results for kernels for Vertex Cover parameterized by the size of $\mathcal{C}$-modulators for different graph classes $\mathcal{C}$, we have attempted to unify and generalize existing results using blocking sets.

First of all, we showed in Chapter 4 that bounded minimal blocking set size in $\mathcal{C}$ is necessary but not sufficient to get a polynomial kernel for Vertex Cover when parameterized by the size of a modulator $X$ to a robust (or at least union-closed) class $\mathcal{C}$. We then showed that bounded minimal blocking set size suffices to efficiently reduce the number of components of $G - X$ assuming that $\mathcal{C}$ is robust (so deletion of components lets $G - X$ stay in $\mathcal{C}$) and that we can efficiently compute optimum vertex covers and test blocking sets in graphs of $\mathcal{C}$. The obtained bound of $\mathcal{O}(|X|^{\beta_\mathcal{C}})$ components is likely optimal because it matches the size of the lower bound proved earlier, which requires only components of constant size. Thus, two key ingredients for the existence of a polynomial kernel for Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator are that Vertex Cover is polynomial-time solvable on $\mathcal{C}$ and that $\mathcal{C}$ has bounded minimal blocking set size.

Starting from the work of Fomin and Strømme [FS16] we showed that $d$-quasi-forests have bounded minimal blocking set size, and that this bound is tight (Section 5.2). Afterwards, we also established tight bounds for the size of minimal blocking sets in $d$-quasi-bipartite graphs and $d$-quasi-integral graphs. In a next step, motivated by the bounds for minimal blocking sets that Bougeret and Sau [BS19] obtained relative to bounded treedepth, we proved in Section 5.3 bounds for the minimal blocking set size relative to elimination distances to classes $\mathcal{C}$. We obtain the exact value for all hereditary classes $\mathcal{C}$ and slightly weaker upper bounds for certain non-hereditary classes $\mathcal{C}$.

In Chapter 6 we presented new results for polynomial kernels for Vertex Cover subject to structural parameters. Our polynomial kernel for Vertex Cover parameterized by the size of a modulator to a $d$-quasi-forest shows that bounds on the feedback vertex set size are more meaningful for kernelization than the treewidth of $G-X$ (recalling that there is a lower bound for treewidth of $G-X$ being at most two). By extending our kernelization to work for modulators to ($d$-quasi-bipartite and) $d$-quasi-integral graphs, we encompassed existing kernelizations for parameterization by distance to forests [JB13], distance to max degree two [MRS18] (both previously subsumed by), distance to pseudoforests [FS16], and parameterization above fractional optimum [KW12]. Furthermore, we enabled polynomial kernelization results for Vertex Cover parameterized by the size of a $(\mathcal{C}_{forest}, d)$-modulator or a $(\mathcal{C}_{\text{LP}}, d)$-modulator. Both parameters generalize the parameter size of a $d$-treedepth modulator as well as the parameter size of a $\mathcal{C}$-modulator with $\mathcal{C} = \mathcal{C}_{forest}$ or $\mathcal{C} = \mathcal{C}_{\text{LP}}$, respectively.

**Open problems.** As future work it would be nice to get a similar kernelization result when parameterized by the size of a modulator to bounded elimination distance to the graph class $\mathcal{C}_{2\text{LP}-\text{MM}}$ where $\text{VC} = 2\text{LP} - \text{MM}$ (i.e., minimum vertex cover size equals two times fractional cost minus size of a maximum matching, cf. [GP16]), which relates to the randomized kernelization for the corresponding above guarantee parameterization [Kra18]. This would essentially subsume and generalize all currently known polynomial kernelizations for Vertex Cover (to which we came close with the result for bounded elimination distance to $\mathcal{C}_{\text{LP}}$).

One way to achieve such a result is to bound the size of minimal blocking sets of graphs $G$ with $d = \text{VC}(G) - (2\text{LP}(G) - \text{MM}(G))$, for constant $d$. Since every minimal blocking set of a graph $G$ with $\text{VC}(G) = 2\text{LP}(G) - \text{MM}(G)$ has size at most 3, a possible guess for the largest minimal blocking set size of a graph $G$ with $d = \text{VC}(G) - (2\text{LP}(G) - \text{MM}(G))$ could be $2d + 3$. Furthermore, the clique $K$ of size $2d + 3$ has minimal blocking set size $2d + 3$ and it holds that $\text{VC}(K) - (2\text{LP}(K) - \text{MM}(K)) = 2d + 2 - (2 \cdot \frac{2d+3}{2} - \frac{2d+2}{2}) = d$.

A possible approach to bound the largest minimal blocking set size of such graphs is similar to the proof of Lemma 5.9. Here, given a minimal blocking set $Y$ of a graph $G$ that is contained in class $\mathcal{C}_{2\text{LP}-\text{MM}}$ we reduce $G$ to an $\frac{1}{2}$-quasi-integral graph $G'$ such that $Y$ is also a minimal blocking set of $G'$. The question is whether given a minimal blocking set $Y$ of a graph $G$ with $d = \text{VC}(G) - (2\text{LP}(G) - \text{MM}(G))$ can we reduce $G$ to a $(d + \frac{1}{2})$-quasi-integral (or $f(d)$-quasi-integral) graph $G'$ such that $Y$ is also a minimal blocking set of $G'$. Observe that Reduction Rules 5.1 to 5.3 also hold when $\mathcal{C}$ is the class of graphs where $\text{VC}(G) + d = 2\text{LP}(G) - \text{MM}(G)$. But, we use that $\text{VC}(G) = 2\text{LP}(G) - \text{MM}(G)$ to prove Claim 5.10, i.e., the finial reductions do not work anymore.

It would also be nice to have tight bounds for the maximum size of minimal blocking sets in the non-hereditary case, and to get such bounds with fewest possible technical assumptions. Besides, it would be interesting whether there are matching upper bounds for kernelization, e.g., whether the kernelization of Jansen and Bodlaender [JB13] for modulators to forests can be improved to size $\mathcal{O}(|X|^2)$.

Furthermore, we showed that to obtain polynomial kernels for Vertex Cover parameterized by the size of a modulator $X$ to graph class $\mathcal{C}$ it is necessary that Vertex Cover is polynomial time solvable on $\mathcal{C}$ and that $\beta_{\mathcal{C}}$ is bounded. More precisely, we showed that bounded minimal blocking set size implies that we can bound the number of connected components. A natural follow-up question is, what is necessary to reduce the size of a component components in $G - X$? Do we need an extra assumption for $\mathcal{C}$ or are these two assumptions sufficient?

Overall, as a result of this Part, there is now a solid understanding of which parameterizations of Vertex Cover lead to fixed-parameter tractability or existence of a polynomial kernelization. This leads to the following picture for structural parameters:

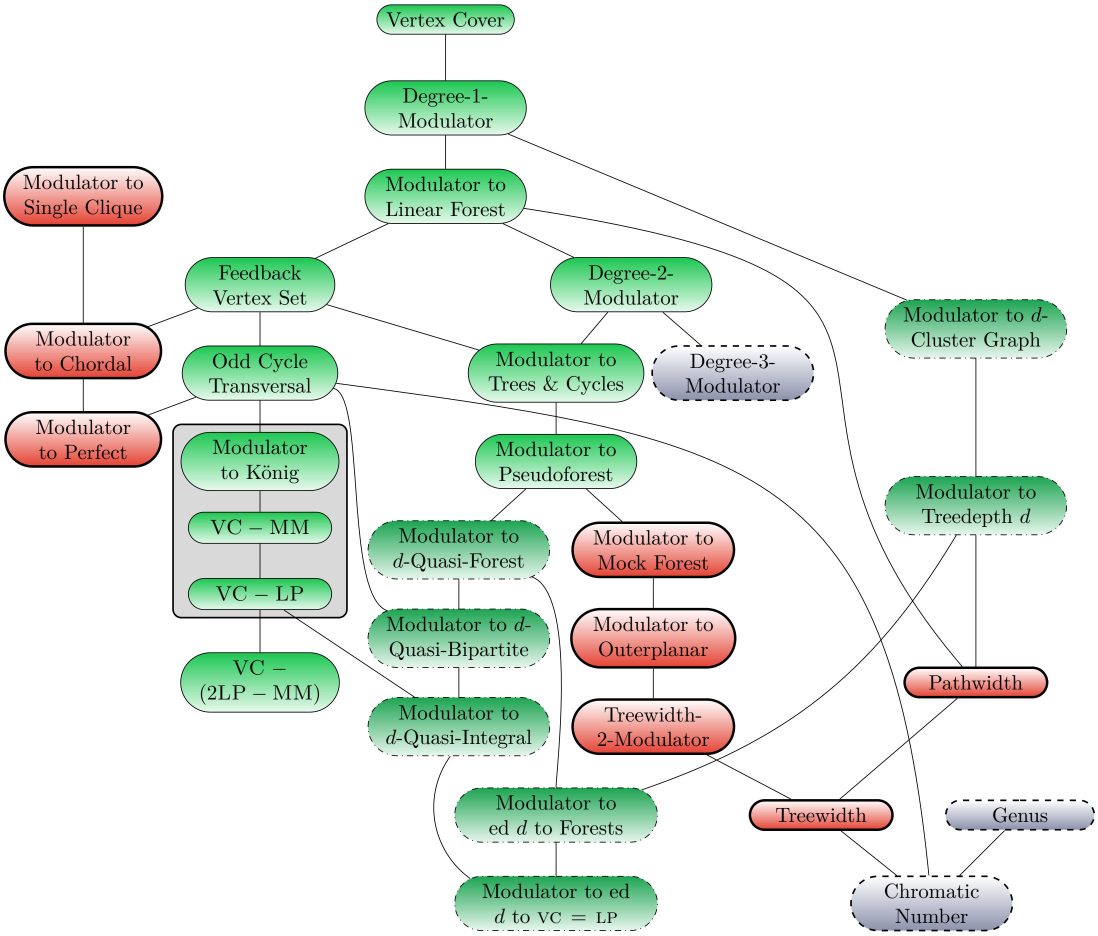



Figure 7.1.: Parameter hierarchy for Vertex Cover, where we assume that the modulator is given in the input. The shadings indicated that the parameterization is either para-NP-complete, FPT but conditionally lacking a polynomial kernel, FPT with a polynomial kernel for constant $d$, or FPT with polynomial kernel. A line between two parameters indicates that the lower parameter can be bounded in a function of the higher parameter. The three parameters that are grouped are equivalent up to constant factors.

# Part III.

# Edge Dominating Set

**– The Stubborn Problem –**

# CHAPTER 8 INTRODUCTION TO EDGE DOMINATING SET

## 8.1. Introduction

In the Edge Dominating Set problem (EDS) we are given a graph $G = (V, E)$, an integer $k$, and need to determine whether there is a set $F \subseteq E$ of at most $k$ edges that are incident with all (other) edges of $G$. It is one of the earliest-known NP-complete problems highlighted by Garey and Johnson [GJ79]. It is well known that Edge Dominating Set is related to matching problems: Clearly, every maximal matching is an edge dominating set. In addition, given a (minimal) edge dominating set $F$ one can construct a maximal matching of size at most $|F|$ in linear time as follows. Take any edge $e = \{x, y\} \in F$ that shares an endpoint with another edge in $F$. We assume, without loss of generality, that $x$ is also endpoint of another edge in $F$. First, if $y$ is also endpoint of another edge in $F$ or if every neighbor of $y$ is an endpoint of an edge in $F$, then $F - e$ is also an edge dominating set of $G$. Thus, we can delete $e$ from $F$ and we decrease the number of edges that share an endpoint. Second, if $y$ has a neighbor $u$ that is not an endpoint of any edge in $F$, then we replace edge $e$ in $F$ by edge $e' = \{y, u\}$. The resulting set $F' = F \setminus \{e\} \cup \{e'\}$ is an edge dominating set of $G$ where less edges share an endpoint. Hence, finding an edge dominating set of size at most $k$ is equivalent to finding a maximal matching of size at most $k$. This implies that Edge Dominating Set admits a simple 2-approximation by taking any maximal matching of $G$.

Besides being related to matching problems, Edge Dominating Set is also related to the Vertex Cover problem (cf. [Fer06]). Indeed, the endpoints of any edge dominating set of size at most $k$ are always a vertex cover of size at most $2k$. Conversely, given any vertex cover $S \subseteq V$ one can find an edge dominating set of size between $\frac{|S|}{2}$ and $|S|$ as follows. Compute a maximum matching $M$ of the graph induced by $S$. A feasible edge dominating set then consists of $M$ together with a single edge incident with each vertex of $S$ that is exposed by $M$. Clearly, the set of endpoints of selected edges contains $S$, thereby dominating all edges, and the size is as claimed. Unfortunately, taking just any (approximate) vertex cover or even a minimum one will not always succeed in finding a minimum edge dominating set. To find an edge dominating set of size at most $k$ (if one exists), it does suffice, however, to try *all minimal* vertex covers of size at most $2k$. This and more elaborate branching strategies (still involving to choose first the endpoints of the edge dominating set) have led to the currently fastest parameterized algorithms for EDGE DOMINATING SET [IN16].

The mentioned relation between EDGE DOMINATING SET and VERTEX COVER implies that EDGE DOMINATING SET parameterized by the solution size $k$ is fixed-parameter tractable because we can enumerate all minimal vertex covers of size at most $2k$ in $\mathcal{O}^*(2^{2k})$ time. Fernau [Fer06] was the first who improved this running time to $\mathcal{O}^*(2.6181^k)$ by having a closer look at the search tree. Later, Fomin et al. [FGSS09] used a combination of branching and treewidth techniques to obtain an algorithm for EDGE DOMINATING SET parameterized by the solution size $k$ that runs in time $\mathcal{O}^*(2.4181^k)$. Using Measure & Conquer, Binkele-Raible and Fernau [BF12] improved the running time to $\mathcal{O}^*(2.3819^k)$. In 2011, Xiao et al. [XKP13] gave an $\mathcal{O}^*(2.3147^k)$ time algorithm for EDGE DOMINATING SET by again enumerating minimal vertex covers of size at most $2k$. Iwaide and Nagamochi [IN16] improved this to the currently best known algorithm which runs in time $\mathcal{O}^*(2.2351^k)$ and also enumerates minimal vertex covers.

Regarding kernelization, we find further similarities between EDGE DOMINATING SET and VERTEX COVER. Prieto [Pri05] showed a kernelization to $4k^2 + 8k$ vertices for the standard parameterization by the solution size $k$ using a crown-like structure. One year later, Fernau [Fer06] observed that enumerating minimal vertex covers of size at most $2k$ can also be used to obtain a kernel for EDGE DOMINATING SET parameterized by $k$ using the VERTEX COVER kernel due to Buss and Goldsmith [BG93]. This leads to a kernel with at most $8k^2$ vertices. Furthermore, Fernau [Fer06] mentioned that it is even possible to reduce to $4k^2$ vertices by dealing with degree one vertices. This was improved to $2k^2 + 2k$ vertices and $\mathcal{O}(k^3)$ edges by Xiao et al. [XKP13] and further tweaked by Hagerup [Hag12] to $\max\{\frac{1}{2}k^2 + \frac{7}{2}k, 6k\}$ vertices and at most $\frac{8}{27}k^3 + \mathcal{O}(k^2)$ edges. Both algorithms use the fact that high degree vertices must be an endpoint of at least one solution edge. Recently, Gao and Gao [GG18] showed that one can reduce to $(d + 3)k$ vertices when the input graph has maximum degree $d$.

**Related work.** The parameterized complexity of EDGE DOMINATING SET has been studied in a number of papers [WCFC09, Xia10a, XN13, EMPX15]. Structural parameters were studied, e.g., by Escoffier et al. [EMPX15] who obtained an $\mathcal{O}^*(1.821^\ell)$ time algorithm where $\ell$ is the vertex cover size of the input graph, and by Kobler and Rotics [KR03] who gave a polynomial-time algorithm for graphs of bounded clique-width. It is easy to see we can express EDGE DOMINATING SET as a $\mathsf{MSO}_2$ formula which implies that EDGE DOMINATING SET is fixed-parameter tractable when parameterized by the treewidth [ALS91]. EDGE DOMINATING SET has also been studied from the perspective of approximation [FN02, CC06, CLL09a, SV12], enumeration [KLMN12, GHKV15, KLM$^+$15], and exact exponential-time algorithms [RSS07, vRB12, XN14].

## 8.2. Structural Parameter for Edge Dominating Set

As in the Vertex Cover problem, the drawback of choosing the solution size $k$ as the parameter is that $k$ is large on many types of easy instances, i.e., graphs of degree at most one, linear forests or forests. Beside that, the size of a minimum edge dominating set is at least equal to half the size of a maximum matching of $G$, which is why the solution size $k$ must have at least half the size of a maximum matching for instances not to be trivially negative. We have seen in the previous part that this has been addressed for Vertex Cover by turning to *structural parameters* that are independent of the solution size.

Having in mind the connection between Vertex Cover and Edge Dominating Set, it is natural to ask which of these results carry over to Edge Dominating Set, both regarding fast parameterized algorithms and the existence of polynomial kernelizations. Edge Dominating Set and Vertex Cover have very similar results for their standard parameterizations but does this remain true for structural parameters, which are necessary to avoid large parameter values? As for Vertex Cover, there exists no polynomial kernel for Edge Dominating Set parameterized by width-parameters like pathwidth, treewidth, or treedepth unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ and the polynomial hierarchy collapses (cf. [BDFH09, BJK14]).

Now, let us consider Edge Dominating Set parameterized by the size of a modulator to a tractable class $\mathcal{C}$. It was shown by Yannakakis and Gavril [YG80] that Edge Dominating Set remains $\mathsf{NP}$-complete on planar graphs with maximum degree three as well as bipartite graphs with maximum degree three. Horton and Kilakos [HK93] showed that the problem is even $\mathsf{NP}$-hard on planar bipartite graphs. Thus, in contrast to Vertex Cover, which admits a randomized polynomial kernel when parameterized by the size of an odd cycle transversal, Edge Dominating Set is para-$\mathsf{NP}$-complete when parameterized by the size of an odd cycle transversal. This raises the question whether Edge Dominating Set parameterized by the size of a modulator to a (linear) forest is even fixed-parameter tractable? The answer to the question is yes, because the treewidth of a graph is at most the size of a modulator to a (linear) forest plus one. Since Edge Dominating Set parameterized by the treewidth is fixed-parameter tractable (as mentioned above) it follows that Edge Dominating Set parameterized by the size of a modulator to a (linear) forest is fixed-parameter tractable.

Unfortunately, we will show in Chapter 9 that Edge Dominating Set parameterized by the size of a modulator to a linear forest does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. More precisely, we will show that Edge Dominating Set parameterized by the deletion distance to a disjoint union of paths of length two does not have a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ (Section 9.2). In Chapter 10, we will show implicitly that we can extend this lower bound when parameterized by the size of a modulator to a cluster graph where each clique has size tree. Observe that this rules out polynomial kernels for Edge Dominating Set parameterized by the size of a modulator to all hereditary graph classes that contain graphs with a degree three vertex.

As Vertex Cover, Edge Dominating Set parameterized by deletion distance to a single clique is fixed-parameter tractable[1] but does not have a polynomial kernel. One can show this by a simple polynomial parameter transformation from Vertex Cover parameterized by the deletion distance to single clique to Edge Dominating Set parameterized by the same parameter.

**Theorem 8.1.** Edge Dominating Set *parameterized by the deletion distance to a single clique does not have a polynomial kernel, unless* NP $\subseteq$ coNP/poly*.*

**Proof.** Let $(G, k, X)$ be an instance of Vertex Cover parameterized by the distance to a single clique, i.e., $G - X$ is a single clique. We construct an instance $(G', k, X)$ of Edge Dominating Set parameterized by the distance to a single clique by adding a set $U = \{u_1, u_2, \dots, u_k\}$ of $k$ vertices to $G$ and by connecting each vertex $u \in U$ to all other vertices of $U$ and $G$; we denote the resulting graph by $G'$. It holds that $G' - X$ is a clique because all new added vertices are adjacent to all other vertices of $G'$, and because $G - X$ is a clique. It remains to show that $G$ has a vertex cover of size at most $k$ if and only if $G'$ has an edge dominating set of size at most $k$.

Let $S \subseteq V(G)$ be a vertex cover of $G$ of size at most $k$. We can assume, without loss of generality, that $S$ contains $k$ vertices; otherwise we add vertices to $S$ until $S$ contains $k$ vertices. Let $S = \{s_1, s_2, \dots, s_k\}$. We will show that $F = \{\{s_i, u_i\} \mid 1 \leq i \leq k\}$ is an edge dominating set of $G'$. Assume for contradiction that there exists an edge $e = \{x, y\} \in E(G')$ such that $\{x, y\} \cap V(F) = \emptyset$. This implies that $x$ and $y$ are vertices of $V(G)$ because the set $U$ of new added vertices is contained in $V(F)$. But, $S \subseteq V(F)$ which contradicts the assumption that $S$ is a vertex cover of $G$. Hence, $F$ is an edge dominating set of $G'$ of size $k$.

For the other direction, we assume that there exists an edge dominating set $F$ of size at most $k$ in $G'$. If there exists a vertex $u \in U$ that is not endpoint of an edge in $F$ then every vertex in $V(G)$ as well as every vertex in $U \setminus \{u\}$ must be endpoint of an edge in $F$ because $u$ is adjacent to all vertices of $V(G') \setminus \{u\}$. This implies that $G$ contains at most $k+1$ vertices since $V(F) = V(G') \setminus \{u\} = V(G) \cup U \setminus \{u\}$ and since $F$ contains at most $k$ edges. Thus, $S = V(G) \setminus \{v\}$ for an arbitrary vertex $v \in V(G)$ is a vertex cover of $G$ of size at most $k$. Now, we assume that every vertex of $U$ is endpoint of an edge in $F$, i.e., $U \subseteq V(F)$. Let $S = V(F) \cap V(G)$ be the intersection of $V(F)$ with the vertices of $G$. It holds that $S$ contains at most $k$ vertices because $V(F)$ contains at most $2k$ vertices and because the set $U$, which contains $k$ vertices, is contained in $V(F)$. Furthermore, the set $S$ is a vertex cover of $G$ because $V(F)$ is a vertex cover of $G'$. This shows that $S$ is a vertex cover of size at most $k$ in $G$. This concludes the proof, because Vertex Cover parameterized by the size of a modulator to a single clique has no polynomial kernel, unless NP $\subseteq$ coNP/poly [BJK14]. ■

[1] One can easily observe that such a graph $G$ has at most $(|V(G) \setminus X| + 1) \cdot 2^{|X|}$ minimal vertex covers which can be computed in $\mathcal{O}^*(2^{|X|})$ time which implies that this problem is FPT.

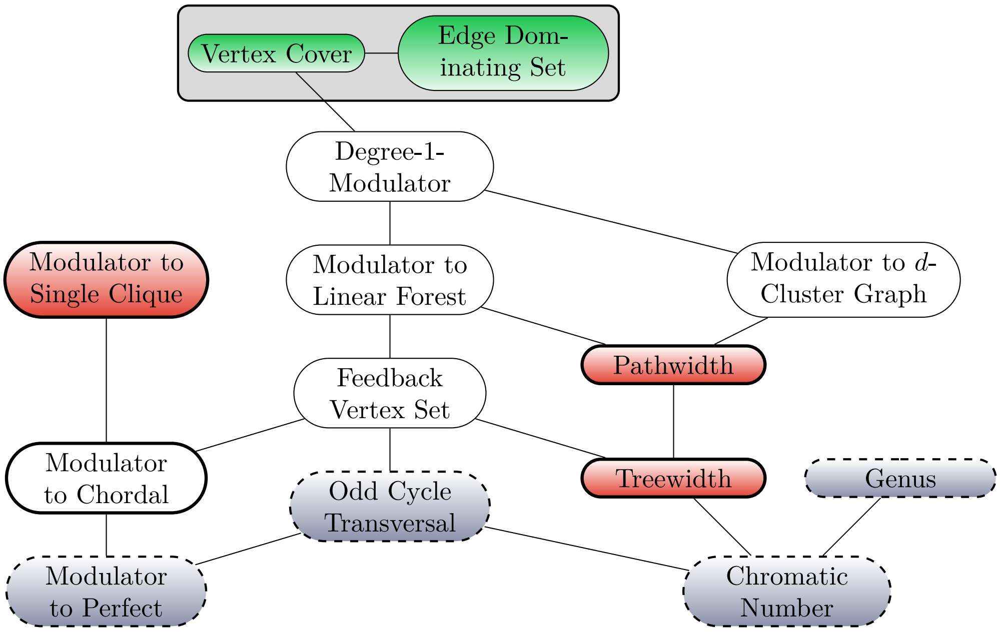



Figure 8.1.: Parameter hierarchy for Edge Dominating Set, where we assume that the modulator is given in the input. The shadings indicated that the parameterization is either para-NP-complete, FPT but conditionally lacking a polynomial kernel, FPT with polynomial kernel, FPT but we do not know whether there exists a polynomial kernel, or unkown whether the problem is NP-hard for constant parameter value. A line between two parameters indicates that the lower parameter can be bounded in a function of the higher parameter. The two parameters that are grouped are equivalent up to constant factors.

So far, this leads to the parameter hierarchy of Figure 8.1. Now, we consider Edge Dominating Set parameterized by a conditional lower bound. A natural lower bound is half the size of a maximum matching. Thus, $\ell = k - \frac{1}{2}\text{MM}$ is a natural lower bound parameter for Edge Dominating Set, where $k$ is the solution size. How does this parameter relate to other parameters? Is it, for example, smaller than the size of an odd cycle transversal?

Consider the four graphs $G_1$, $G_2$, $G_3$ and $G_4$ of Figure 8.2. Graph $G_1$ consists of a clique $K_{2d}$ of size $2d$ and an independent set $I$ of size $2d$ where every edge of the independent set $I$ is adjacent to every vertex of the clique $K_{2d}$. It holds that $\text{EDS}(G_1) - \frac{1}{2}\text{MM}(G_1) = d - \frac{1}{2} \cdot 2d = 0$, but the size of an odd cycle transversal is $2d - 1$, the chromatic number is $2d + 1$ and the size of a modulator to a $c$-cluster graph, for any constant $c$, is $2d$. Similar, graph $G_2$ consists of $2d$ cycles of length five with one attached edge, where the $2d$ vertices that are attached to the additional edge induce a path and it holds that $\text{EDS}(G_2) - \frac{1}{2}\text{MM}(G_2) = \frac{3}{2}d - \frac{1}{2} \cdot 3d = 0$ whereas the size of

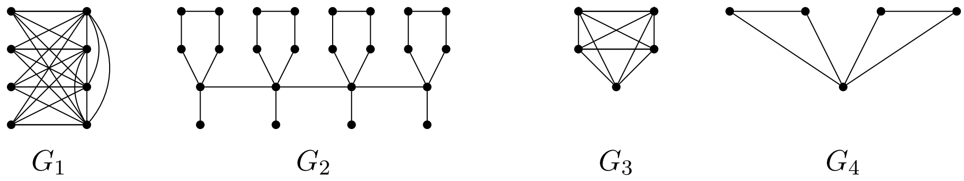



Figure 8.2.: Graphs for $d = 2$

a modulator to a perfect graph is $2d$. This shows that there exist graphs where the parameter $\ell = k - \frac{1}{2}\text{MM}$ is zero and the structural parameters of Figure 8.1 are arbitrary large. Hence, we cannot bound the size of the parameter $\ell = k - \frac{1}{2}\text{MM}$ by any of these parameters.

Conversely, consider the graph $G_3$ which is a clique on $2d+1$ vertices. It holds that $\text{EDS}(G_3) - \frac{1}{2}\text{MM}(G_3) = d - \frac{1}{2}d = \frac{1}{2}d$. But, the size of a modulator to a single clique is zero. Finally, consider the graph $G_4$ which is the disjoint union of $d$ edges whose endpoints are connected to one additional vertex. Again, $\text{EDS}(G_4) - \frac{1}{2}\text{MM}(G_4) = d - \frac{1}{2} \cdot d = \frac{d}{2}$, but the size of a degree-1-modulator and the size of a modulator to a $c$-cluster graph, for any constant $c > 1$, are one. Furthermore, the genus is zero. Thus, there exist graphs where the parameter $\ell = k - \frac{1}{2}\text{MM}$ is arbitrary large, but the structural parameter of Figure 8.1, except the solution size $k$, have constant size. Overall, this shows that the parameter $\ell = k - \frac{1}{2}\text{MM}$ is incomparable to the other parameters that we consider in Figure 8.1, except the solution size $k$ which is obviously an upper bound of $\ell$.

## 8.3. Preliminaries

Let $\mathcal{H}$ be a set of graphs. We say that a graph $G$ is an *$\mathcal{H}$-component graph* if each connected component of $G$ is isomorphic to some graph in $\mathcal{H}$. Clearly, disconnected graphs in $\mathcal{H}$ do not affect which graphs $G$ are $\mathcal{H}$-component graphs and, thus, our proofs need only consider the connected graphs $H \in \mathcal{H}$. We write $H$-component graph rather than $\{H\}$-component graph for single (connected) graphs $H$.

All our composition-based proofs in this part reduce from the NP-hard MULTICOLORED CLIQUE problem. Therein we are given a graph $G = (V, E)$, an integer $k$, a partition of $V$ into $k$ sets $V_1, \ldots, V_k$ of equal size and we need to determine whether there is a clique $X$ of size $k$ in $G$ that contains exactly one vertex from each set $V_i$. Such a set $X$ is called a *multicolored $k$-clique*.

**Overview of this Part.** In this part we study EDGE DOMINATING SET under different structural parameters. Our work appears to be the first to study the existence of polynomial kernels for EDGE DOMINATING SET subject to structural parameters, except some lower bounds, e.g., for parameter treewidth, as mentioned above.

In Chapter 9 we show that EDGE DOMINATING SET parameterized by the size of a modulator to a $P_3$-component graph does not have a polynomial kernel unless

$\mathsf{NP} \subseteq \mathsf{coNP/poly}$ (Section 9.2). Afterwards, we show that EDGE DOMINATING SET parameterized by the size of a degree-1-modulator has a polynomial kernel (Section 9.3). Having a closer look at the kernel for EDGE DOMINATING SET parameterized by the size of a degree-1-modulator, one can observe that similar reduction rules can be obtained for a kernel for EDGE DOMINATING SET parameterized by the size of a modulator to a $P_5$-component graph (Section 9.4). Finally, we show in Section 9.5 that EDGE DOMINATING SET parameterized by $\ell = k - \frac{1}{2}\text{MM}$ is para-NP-complete. Some of the results are summarized in Figure 8.3.

Motivated by the results of Chapter 9, namely that even constant-size components in $G - X$ behave in a nontrivial way regarding kernelization by $|X|$, we consider EDGE DOMINATING SET parameterized by the size of a modulator to an $\mathcal{H}$-component graph for a given finite set $\mathcal{H}$ of graphs (Chapter 10). We classify for each finite set $\mathcal{H}$ of graphs whether EDGE DOMINATING SET parameterized by the size of a modulator to an $\mathcal{H}$-component has a polynomial kernel or not. Therefore, we generalize the lower bound construction for $P_3$-component graphs (Section 10.3) and the kernel for $P_5$-component graphs (Section 10.4).

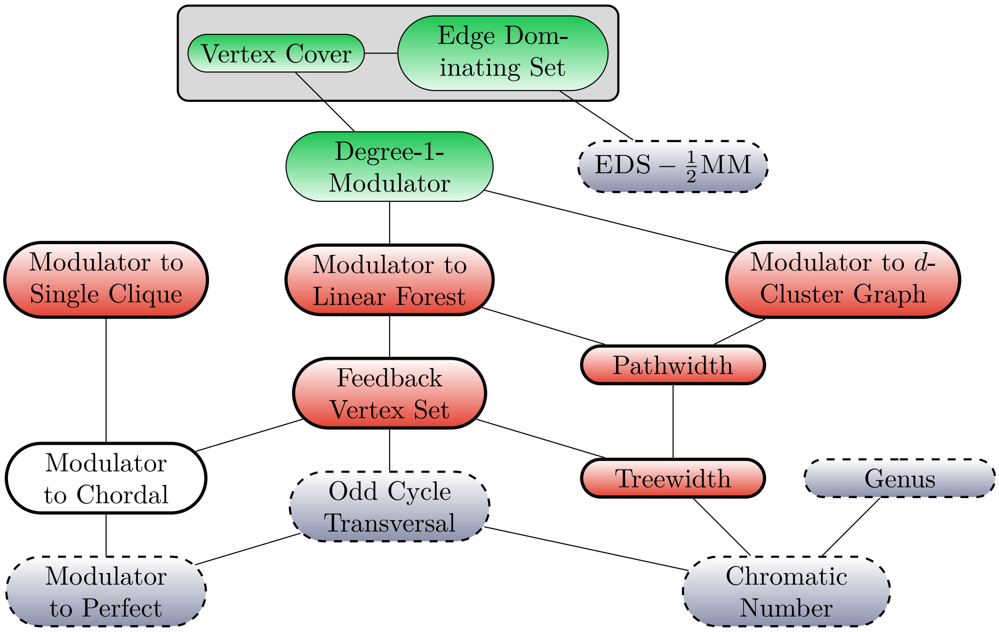



Figure 8.3.: Parameter hierarchy for EDGE DOMINATING SET, where we assume that the modulator is given in the input. The shadings indicated that the parameterization is either para-NP-complete, FPT but conditionally a lacking polynomial kernel, FPT with polynomial kernel, or unkown whether the problem is NP-hard for constant parameter value. A line between two parameters indicates that the lower parameter can be bounded in a function of the higher parameter. The two parameters that are grouped are equivalent up to constant factors.

# CHAPTER 9 PARAMETERIZATION BY STRUCTURAL PARAMETERS

## 9.1. Introduction

Motivated by the number of positive results for VERTEX COVER parameterized by structural parameters we would like to know whether some of these results carry over to the related but somewhat more involved EDGE DOMINATING SET problem. For kernelization subject to the size of a modulator to some tractable class $\mathcal{C}$ there is bad news: Even if $\mathcal{C}$ contains only the disjoint unions of paths of length two (consisting of three vertices each) we show that there is no polynomial kernelization for parameterization by $|X|$ with $G - X \in \mathcal{C}$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ and the polynomial hierarchy collapses (Section 9.2). The same is true when $\mathcal{C}$ contains at least all disjoint unions of triangles.[1] Thus, for the usual program of studying modulators to well-known *hereditary* graph classes $\mathcal{C}$ there is essentially nothing left to do because the only permissible connected components would have one or two vertices. This very modest case actually admits a polynomial kernelization (Section 9.3).

Somewhat surprisingly, we then show that EDGE DOMINATING SET parameterized by the size of a modulator to a $P_5$-component graph has a polynomial kernel (Section 9.4). That said, considering only hereditary graph classes would ignore an interesting landscape of positive and negative results that can be obtained by permitting certain forms of connected components in $G - X$ but not necessarily all induced subgraphs thereof.[2]

Regarding parameterization above lower bounds, we prove that it is $\mathsf{NP}$-hard to determine whether a graph $G$ has an edge dominating set of size equal to the lower bound of half the size of a maximum matching. This rules out any positive results for parameter $\ell = k - \frac{1}{2}\textsc{mm}$ (Section 9.5).

[1] We do not show this result explicitly. However, it follows from Theorem 10.4 in Chapter 10.

[2] We consider such parameters for EDGE DOMINATING SET in Chapter 10.

## 9.2. Modulator to a $P_3$-Component Graph

In this section, we show that Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. This rules out polynomial kernels for a large number of interesting parameters like feedback vertex set size or size of a modulator to a linear forest. To prove this we give a cross-composition from Multicolored Clique.

**Theorem 9.1.** Edge Dominating Set *parameterized by the size of a modulator to a $P_3$-component graph (and thus also parameterized by the size of a modulator to a linear forest) does not admit a polynomial kernel unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

**Proof.** To prove the theorem we give a cross-composition from the NP-hard Multicolored Clique problem to Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph. Input instances are of the form $(G_i, k_i)$ where $G_i$ comes with a partition of the vertex set into $k$ color classes. (Since the color classes are of equal size it holds that $k \leq |V(G_i)|$.) For the polynomial equivalence relation $\mathcal{R}$ we take the relation that puts two instances $(G_1, k_1)$, $(G_2, k_2)$ of Multicolored Clique in the same equivalence class if $k_1 = k_2$ and $|V(G_1)| = |V(G_2)|$. It is easy to check that $\mathcal{R}$ is a polynomial equivalence relation. (Instances with size at most $N$ have at most $N$ vertices. Thus, we get at most $N^2$ classes for instances of size at most $N$.)

Let a sequence of instances $I_i = (G_i, k)_{i=1}^{t}$ of Multicolored Clique be given that are equivalent under $\mathcal{R}$. We identify the color classes of the input graphs so that all graphs have the same vertex set $V$ and the same color classes $V_1, V_2, \ldots, V_k$. Let $n := |V_i|$ be the number of vertices of each color class; thus, each instance has $|V| = n \cdot k$ vertices. We assume, without loss of generality, that every instance has at least one edge in $E(V_p, V_q)$ for all $1 \leq p < q \leq k$; otherwise, this instance would be a trivial no instance and we can delete it. Furthermore, we can assume, without loss of generality, that $t = 2^s$ for an integer $s$, since we may copy some instances if needed (while at most doubling the number of instances and increasing $\log t$ by less than one).

Now, we construct an instance $(G', k', X')$ of Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph, where the size of $X'$ is polynomially bounded in $n + k + s$ (see Figure 9.1 for an illustration). We add a set $V$ consisting of $k \cdot n$ vertices to graph $G'$ which represents the vertices of the $t$ instances. The set $V$ is partitioned into the $k$ color classes $V_1, V_2, \ldots, V_k$. To choose which vertices are contained in a clique of size $k$, we add a set $T = \{t_1, t_2, \ldots, t_k\}$ and a set $T' = \{t'_1, t'_2, \ldots, t'_k\}$, each of size $k$, to $G'$. We make $t_j \in T$, with $j \in [k]$, adjacent to all vertices in $V_j$ and to vertex $t'_j \in T'$. Next, we add two sets $Z$, $Z'$, each of size $s$, and a set $W$ of size $2s$ to $G'$ and add edges to $G'$ such that each vertex in $Z$ has exactly one private neighbor in $Z'$ and is adjacent to all vertices in $W$. The set $W$ contains $\binom{2s}{s} \geq 2^s$ different subsets of size $s$. For each instance $(G_i, k)$, with $i \in [t]$, we pick a different subset of size $s$ of $W$ and denote it by $W(i)$. For all $1 \leq p < q \leq k$ we add a vertex $s_{p,q}$ and a vertex $s'_{p,q}$ to $G'$; these will correspond to edge sets $E(V_p, V_q)$. Let $S = \{s_{p,q} \mid 1 \leq p < q \leq k\}$ and $S' = \{s'_{p,q} \mid 1 \leq p < q \leq k\}$. We make vertex $s_{p,q}$

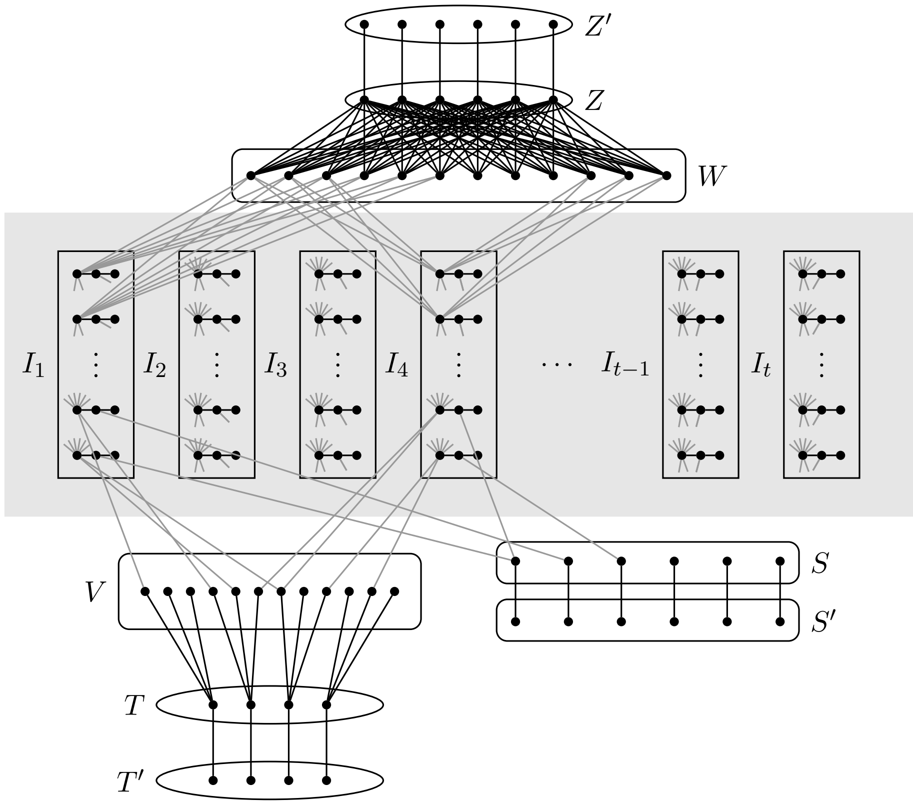



Figure 9.1.: Construction of $G'$ with $k = 4$, where $X' = W \cup Z \cup Z' \cup V \cup T \cup T' \cup S \cup S'$.

adjacent to vertex $s'_{p,q}$ for all $1 \leq p < q \leq k$. For each graph $G_i$, for $i \in [t]$, we add $|E(G_i)|$ paths of length two to the graph $G'$; every $P_3$ represents exactly one edge of the graph $G_i$. Let $P_i^e = u_{i,1}^e u_i^e u_{i,2}^e$ denote the path of instance $i \in [t]$ that represents edge $e \in E(G_i)$. Finally, we make vertices in $P_i^e$, with $i \in [t]$ and $e \in E(G_i)$, adjacent to vertices in the sets $W$, $V$, and $S$ as follows: We make vertex $u_{i,1}^e$ of path $P_i^e$, with $i \in [t]$, which represents edge $e = \{x, y\} \in E(G_i)$ adjacent to the vertices $x, y$ in $V$ and to all vertices in the set $W(i) \subseteq W$. Additionally, we make vertex $u_i^e$ adjacent to vertex $s_{p,q}$ where $1 \leq p < q \leq k$ such that $e \in E(V_p, V_q)$.

The set $X'$ is defined to contain all vertices that do not participate in the paths $P_i^e$, i.e., $X' = W \cup Z \cup Z' \cup V \cup T \cup T' \cup S \cup S'$. Clearly, $G - X'$ is a $P_3$-component graph and $|X'| = 4s + k \cdot n + 2k + 2 \cdot \binom{k}{2}$. Let $k' = k + s + \sum_{i=1}^{t} |E(G_i)|$. Note that the size of $k'$ can depend linearly on the number of instances, because our parameter is the size of $X'$, which is polynomially bounded in $n + s$, as $k \leq n$. We return the instance $(G', k', X')$; clearly, this instance can be generated in polynomial time.

Now, we have to show that $(G', k', X')$ is a yes-instance of Edge Dominating Set if and only if there exists an $i^* \in [t]$ such that $(G_{i^*}, k)$ is a yes-instance of Multicolored Clique.

($\Rightarrow$:) Assume first that $(G', k', X')$ is a yes-instance of Edge Dominating Set and that there exists an edge dominating set $F$ of size at most $k'$ in $G'$. We can always pick $F$ such that it fulfills the following properties (most hold for all solutions of size at most $k'$):

1. The vertex sets $S$, $T$, and $Z$ must be subsets of $V(F)$: E.g., for each edge $\{z, z'\}$ with $z \in Z$ and $z' \in Z'$ the set $V(F)$ must contain $z$ or $z'$; if it contains $z'$ then $\{z, z'\} \in F$ as it is the only edge incident with $z'$; either way we get $z \in V(F)$. The same applies for $S$ and $S'$, and for $T$ and $T'$.

2. Because $S, T, Z \subseteq V(F)$ but $S \cup T \cup Z$ is an independent set, the set $F$ must contain at least $|S|$ edges incident with $S$, $|T|$ edges incident with $T$, and $|Z|$ edges incident with $Z$. By straightforward replacement arguments we may assume that $F$ contains exactly the following edges incident with $S \cup T \cup Z$: $|T|$ edges between $T$ and $V$, $|Z|$ edges between $Z$ and $W$, and $|S|$ edges between $S$ and middle vertices $u_i^e$ of $P_3$'s in $G' - X'$. Furthermore, we can assume that these edges are a matching, because no color class is empty, no edge set $E(V_p, V_q)$ is empty, and $Z$ is adjacent to all vertices in $W$.

3. For each path $P_i^e = u_{i,1}^e u_i^e u_{i,2}^e$, which represents the edge $e$ of instance $(G_i, k)$, at least vertex $u_i^e$ must be an endpoint of an edge in $F$: Indeed, to cover the edge $\{u_i^e, u_{i,2}^e\}$ one of its two vertices must be in $V(F)$. Similar to property 1 above, if $u_{i,2}^e \in V(F)$ then $F$ must contain its sole incident edge $\{u_i^e, u_{i,2}^e\}$ and, hence, $u_i^e \in V(F)$.

4. An edge in $F$ cannot have its endpoints in two different $P_3$'s of $G' - X'$ because no such edges exist.

Let $F_T = F \cap E(T, V)$, let $F_Z = F \cap E(Z, W)$, let $F_S = F \cap E(S, \{u_i^e \mid i \in [t], e \in E(G_i)\})$, and let $F_R = F \setminus (F_T \cup F_Z \cup F_S)$. Hence, due to properties 1 and 2, we have

$$|F_R| \leq k' - |F_T| - |F_Z| - |F_S| \leq \sum_{i=1}^{t} |E(G_i)| - \binom{k}{2}.$$

By property 3, all vertices $u_i^e$ are endpoints of edges in $F$. Among $F_T \cup F_Z \cup F_S$ this can only be true for the $|S| = \binom{k}{2}$ edges in $F_S$. Since there are exactly $\sum_{i=1}^{t} |E(G_i)|$ vertices $u_i^e$, which is (greater or) equal to $|F_R| + |F_S|$, and there are no edges connecting different such vertices (property 4), each edge in $F_R \cup F_S$ is incident with a private vertex $u_i^e$. This also implies that all edges in $F_R$ have no endpoints in $V \cup W$ as those sets are not adjacent to any vertex $u_i^e$. Thus, in $W$ exactly the $|Z| = s$ endpoints of $F_Z$ are endpoints of $F$. Similarly, in $V$ exactly the $|T| = k$ endpoints of $F_T$ are endpoints of $F$. Let $X \subseteq V$ denote this set of $k$ vertices. Observe that by construction of $G'$ the set $X$ contains exactly one vertex from each color class, because $t_j \in T$, for $j \in [k]$, is only adjacent to vertices of $V_j$.

Now, consider any path $P_i^e = u_{i,1}^e u_i^e u_{i,2}^e$ where $u_i^e$ is an endpoint of an edge $f \in F_S$. Clearly, the other endpoint of $f$ lies in $S$, and, by the above accounting, no other edge of $F$ is incident with $u_{i,1}^e$ or $u_{i,2}^e$. In particular, this implies that all neighbors of $u_{i,1}^e$ in $W$ and $V$ must be endpoints of edges in $F$. If $e = \{x, y\}$ then these neighbors of $u_{i,1}^e$ are the set $W(i) \subseteq W$ and the vertices $x, y \in V$, and, by construction of $G'$, the edge $\{x, y\}$ must exist in $G_i$. Thus, $W(i) \cup \{x, y\} \subseteq V(F)$ which implies that $x, y \in X$.

Repeating this argument for all $|S| = \binom{k}{2}$ paths of this type, we can conclude the following: (1) All paths correspond to the same instance $i^* \in [t]$ because we require $W(i) \subseteq V(F)$, but exactly $|Z| = |W(i^*)| = s$ such vertices are in $V(F)$. (Different values of $i$ would require different sets $W(i)$, exceeding size $s$.) (2) There are $\binom{k}{2}$ edges of $G_{i^*}$ represented by the paths and all their endpoints must be in $X = V \cap V(F)$. Since $|X| = k$, the edges must form a clique of size $k$ on vertex set $X$ in $G_{i^*}$. We already observed above that $X$ contains exactly one vertex per color class, hence, instance $(G_{i^*}, k)$ is a yes-instance of Multicolored Clique, as claimed.

($\Leftarrow$:) For the other direction, assume that for some $i^* \in [t]$ the Multicolored Clique instance $(G_{i^*}, k)$ is a yes-instance. Let $X = \{x_1, x_2, \ldots, x_k\} \subseteq V$ be a multicolored clique of size $k$ in $G_{i^*}$ with $x_j \in V_j$ for $j \in [k]$, let $E'$ be the set of edges of the clique $X$, and let $e_{p,q} = \{x_p, x_q\}$, for $1 \leq p < q \leq k$, be the one edge in $E' \cap E(V_p, V_q)$. We construct an edge dominating set $F$ of $G'$ of size at most $k'$ as follows. First we add the $k$ edges $\{t_j, x_j\}$ for $j \in [k]$ between $T$ and $X \subseteq V$; thus, $T \cup X \subseteq V(F)$. We then add a maximum matching (of size $s$) between $W(i^*) \subseteq W$ and $Z$ to the set $F$. This matching saturates $W(i^*)$ and $Z$ because $|Z| = |W(i^*)| = s$; thus, $W(i^*) \cup Z \subseteq V(F)$. Next, we add the edges $\{u_{i^*}^{e_{p,q}}, s_{p,q}\}$ for all edges $e_{p,q} \in E'$, with $1 \leq p < q \leq k$, to the set $F$; hence $S \subseteq V(F)$. Finally, for all other paths $P_i^e$, with $i \in [t]$, $e \in E(G_i)$, and $i \neq i^*$ or $e \notin E'$, we add the edge $\{u_{i,1}^e, u_i^e\}$ to $F$. (We have thus selected exactly one edge incident with each path of $G' - X'$.) By construction, it holds that $|F| = k + s + \sum_{i=1}^{t} |E(G_i)| = k'$.

It remains to show that $F$ is indeed an edge dominating set of $G'$. To prove this, it suffices to show that $V(G') \setminus V(F)$ is an independent set in $G'$. We already know that $S \cup T \cup W(i^*) \cup X \cup Z \subseteq V(F)$. Moreover, $V(F)$ contains the middle vertex $u_i^e$ for all $P_3$'s in $G' - X'$ and it contains $u_{i,1}^e$ for all $P_3$'s that do not correspond to an edge of the clique $X$ (i.e., with $i \neq i^*$ or with $i = i^*$ but $e \neq e_{p,q}$ for any $1 \leq p < q \leq k$). The sets $S'$, $T'$, and $Z'$ are independent sets whose neighborhoods $S$, $T$, and $Z$ are subsets of $V(F)$. Similarly, all vertices $u_{i,2}^e$ have their single neighbor $u_i^e$ in $V(F)$. Thus, only vertices in $W \setminus W(i^*)$ and $V \setminus X$ could possibly be adjacent to vertices $u_{i^*,1}^{e_{p,q}}$, which correspond to the edges of $G_{i^*}[X]$, in $G' - V(F)$, but this can be easily refuted: Indeed, each $u_{i^*,1}^{e_{p,q}}$ is adjacent only to $x_p$ and $x_q$ in $V$, which are both in $X \subseteq V(F)$, and to the vertices in $W(i^*)$ in $W$, but $W(i^*) \subseteq V(F)$ as well. Thus $V(G') \setminus V(F)$ is an independent set in $G'$ and hence $F$ is an edge dominating set for $G'$ of size at most $k'$. Thus, $(G', k', X')$ is a yes-instance of Edge Dominating Set, which completes the cross-composition.

By Theorem 2.14 the cross-composition from Multicolored Clique to Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graphs implies the claimed lower bound for kernelization. ■

We proved that Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph has no polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. This rules out polynomial kernels for modulators to most frequently studied graph classes (e.g., forests, linear forests, planar graphs). A similar proof establishes the same lower bound for modulators to $K_3$-component graphs. As mentioned in the introduction this rules out polynomial kernels using modulators to essentially all interesting hereditary graph classes. It certainly does completely settle the question for modulators to $\mathcal{H}$-component graphs for all *hereditary* classes $\mathcal{H}$. If $\mathcal{H}$ contains any connected graph with at least three vertices then we get a lower bound; else all connected components have one or two vertices and we will show in the next section that Edge Dominating Set parameterized by the size of a degree-1-modulator has a polynomial kernel.

## 9.3. Degree-1-Modulator

Now, we will show that Edge Dominating Set parameterized by the size of a degree-1-modulator has a polynomial kernel, i.e., the size of a set $X$ such that every vertex in $G - X$ has degree at most 1. Recall that a degree-1-modulator is the same as a modulator to graph where each connected component consists of one or two vertices. This completely classifies the existence of polynomial kernels for parameterization by the size of a $\mathcal{C}$-modulator, where $\mathcal{C}$ is a hereditary graph class.

**Theorem 9.2.** Edge Dominating Set *parameterized by the size of degree-1-modulator $X$ admits a kernel with $\mathcal{O}(|X|^2)$ vertices.*[3]

Theorem 9.2 follows from Lemma 10.15 in Chapter 10. For that reason we only sketch the proof of Theorem 9.2 to give an idea how the polynomial kernel works.

**Proof sketch.** Let $(G, k, X)$ be an instance of Edge Dominating Set parameterized by the size of a degree-1-modulator, i.e., every connected component of $G - X$ consists of at most two vertices. We denote the set of connected components of $G - X$ by $\mathcal{C}$, the set of connected components that contain one vertex by $\mathcal{C}'$, and the set of connected components that contain two vertices by $\mathcal{C}''$.

First, we construct a bipartite graph $G_B$, where one part consists of the set $X$, the other part of one vertex $s_P$ for each connected component $P \in \mathcal{C}''$, and there is an edge between $x \in X$ and $s_P$ with $P \in \mathcal{C}''$ if and only if $P$ is adjacent to $x$. We apply Theorem 2.2 (Hopcroft Karp [HK73]) to graph $G_B$ to obtain either a maximum matching in $G_B$ that saturates $X$ or a set $Y \subseteq X$ such that $|N_{G_B}(Y)| < |Y|$ and such that there exists a maximum matching in $G_B - N_{G_B}[Y]$ that saturates $X \setminus Y$. Let $X_1 = X$ and $X_2 = \emptyset$ if there exists a maximum matching in $G_B$ that saturates $X$, and let $X_1 = X \setminus Y$ and $X_2 = Y$ if we find a set $Y$ with the above properties. Furthermore, let $M$ be a maximum matching in $G_B - N_{G_B}[X_2]$ that saturates $X_1$.

[3]There exists a 3-factor approximation algorithm for finding a degree-1-modulator (cf. [MRS18]).

**Reduction Rule 9.1.** Delete $X_1$ from $G$, i.e., let $G' = G - X_1$, $X' = X \setminus X_1 = X_2$, and $k' = k$.

Claim 9.3. *Reduction Rule 9.1 is safe.*

Proof. Let $F$ be an edge dominating set of size at most $k$ in $G$. We construct an edge dominating set $F'$ of size at most $k' = k$ in $G'$ by deleting every edge $e = \{x, y\} \in F$ if both endpoints of $e$ are contained in $X_1$, or if exactly one endpoint is contained in $X_1$ and the other endpoint is isolated in $G'$, and by replacing every edge $e = \{x, y\} \in F$ with $x \in X_1$ and $y \notin X_1$ by exactly one edge in $\delta_{G'}(y)$ if $\delta_{G'}(y) \neq \emptyset$. It holds that $F'$ has size at most $k = k'$ because we either delete edges in $F$ or we replace one edge by exactly one new edge. Since every vertex in $V(G') \cap V(F)$ is either contained in $V(F')$ or isolated in $G'$ it holds that $F'$ is an edge dominating set in $G'$.

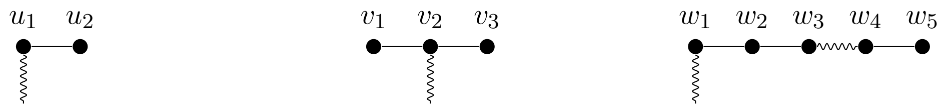
For the other direction, let $F'$ be an edge dominating set of size at most $k'$ in $G'$. For each vertex $x \in X_1$ let $P_x = \{w_1^x, w_2^x\}$ be the path in $\mathcal{C}''$ with $\{x, s_P\} \in M$. Recall, $M$ is the matching in $G_B - N_{G_B}[X_2]$ that saturates $X_1$. It holds that the vertices $w_1^x$, and $w_2^x$ are not adjacent to any vertex in $X' = X_2$; otherwise it would hold that $s_P \in N_{G_B}(X_2)$. Thus, the edge dominating set $F'$ must contain the edge $\{w_1^x, w_2^x\}$. Since at least one of the edges $\{x, w_1^x\}$, $\{x, w_2^x\}$ is contained in $E(G)$, we can obtain an edge dominating set in $G$ by replacing for each vertex $x \in X_1$ the edge $\{w_1^x, w_2^x\}$ in $F'$ by one edge in $E(G) \cap \{\{x, w_1^x\}, \{x, w_2^x\}\}$. We denote the resulting set by $F$. By construction, it holds that $F$ is an edge dominating set of size at most $k$ in $G$. $\square$

In a next step we delete all connected components of $G - X$ that are not adjacent to any vertex of $X$, and decrease $k$ accordingly. This rule is obviously safe.

**Reduction Rule 9.2.** Let $C$ be a connected component of $G - X$ that is not adjacent to any vertex of $X$. Delete $C$ from $G$ and decrease $k$ by $\textsc{eds}(C)$, i.e., $G' = G - C$, $X' = X$, and $k' = k - \textsc{eds}(C)$.

Let $(G, k, X)$ be an instance that is reduced under Reduction Rule 9.1 and Reduction Rule 9.2. Every connected component of size two in $G - X$ must be adjacent to a vertex in $X = X_2$ and hence, corresponds to a vertex in $N_{G_B}(X_2)$. Since $|N_{G_B}(X_2)| < |X_2|$ it follows that $G - X$ contains at most $|X| = |X_2|$ connected components of size two. Thus, it remains to reduce the number of connected components of size one in $G - X$.

We assume that $k - \textsc{eds}(G - X) < |X|$. Otherwise, we can return a trivial solution consisting of an edge dominating set of $G - X$, and one edge in $\delta_G(x)$ for each vertex $x \in X$. Let $X^h \subseteq X$ be the set of vertices in $X$ that are adjacent to at least $|X| + 1$ connected components of $\mathcal{C}'$. Observe that every edge dominating set of size at most $k$ must contain the vertices in $X^h$ as endpoints: If $x \in X^h$ is not an endpoint of an edge dominating set $F$ then there are at least $|X| + 1$ connected components of size one that are contained in $V(F)$. This implies that $|F| \geq \textsc{eds}(G - X) + |X| + 1 > k$ because the other endpoint of these edges (that have one endpoint in one of the at least $|X| + 1$ connected components of size one that are adjacent to $x$) is in $X$. This leads to our final reduction rule:




(a) $P = u_1u_2$ (b) $P' = v_1v_2v_3$ (c) $P'' = w_1w_2w_3w_4w_5$

Figure 9.2.: Comparing $P_2$, $P_3$ and $P_5$

**Reduction Rule 9.3.** Delete all connected components $\{v\}$ in $\mathcal{C}'$ that are only adjacent to vertices in $X^h$, i.e., $N_G(v) \subseteq X^h$, and add for all vertices $x \in X^h$ the vertex $x'$ as well as the edge $\{x, x'\}$ to $G$.

The correctness of the above reduction rule follows from the fact that every edge dominating set of size at most $k$ must contain the set $X^h$. If Reduction Rule 9.3 is no longer applicable then we can bound the number of connected components in $\mathcal{C}'$. Since every connected component in $\mathcal{C}'$ must be adjacent to a vertex in $X \setminus X^h$, and at most $|X|$ of these components are adjacent to the same vertex, it follows that $|\mathcal{C}'| \leq |X|^2$.

Let $(G', k', X')$ be the reduced instance. We showed that $G' - X'$ contains at most $|X|^2$ connected components that consist of one vertex, and at most $|X|$ connected components that consist two vertices. Thus, $G'$ contains at most $3|X| + |X|^2$ vertices. Finally, we show that the reduction can be performed in polynomial time. We apply Reduction Rule 9.1 and Reduction Rule 9.3 at most once, and Reduction Rule 9.2 at most $|V(G)|$ times. Moreover, we can apply each reduction rule in polynomial time, because we can construct the bipartite graph $G_B$, and therefore the set $X_1$, as well as the set $X^h$ in polynomial time. ■

## 9.4. Modulator to a $P_5$-Component Graph

So far we have seen a polynomial kernel for Edge Dominating Set parameterized by the size of a degree-1-modulator and we have seen that Edge Dominating Set has no polynomial kernel when parameterized by the size of a modulator to a $P_3$-component graph. We could stop here and pretend that we understood the source of difficulty with respect to Edge Dominating Set parameterized by the size of a $\mathcal{C}$-modulator. However, having a closer look at Reduction Rule 9.1 one can observe that a similar reduction rule can also be obtained to reduce the number of connected component that are isomorphic to the $P_5$. The only optimum edge dominating set for the path $P = u_1, u_2$ is the edge $\{u_1, u_2\}$. But, instead of using this edge, one could use the local solution $\{u_1, x\}$ or $\{u_2, x\}$ where $x$ is a vertex in the modulator $X$. We used this property to find vertices in the modulator (the set $X_1$) that we can cover without using more edges that one needs for an optimum edge dominating set in $G - X$. Now, for a $P_5$ path $P'' = w_1w_2w_3w_4w_5$, where the optimal edge dominating set has size two, the situation is similar. Here, all vertices but $w_3$ have local solutions that include an edge $\{w_i, x\}$ with $x \in X$; see Figure 9.2. In particular, there are such solutions that also have $w_3$ as an endpoint of the second solution edge on $P''$ for all $w_i \neq w_3$. This is the important difference to the $P_3$ and the reason why we can apply a modified version of Reduction Rule 9.1 when $G - X$ is a disjoint union of $P_5$'s.

Let us try to illustrate the difference between the paths $P_2$, $P_5$ and the path $P_3$ with respect to local optimum solutions. For a $P_3$ path $P' = v_1v_2v_3$ in $G - X$ the optimal edge dominating set (having size one) uses one of the edges $\{v_1, v_2\}$ or $\{v_2, v_3\}$, or any edge $\{v_2, x\}$ with $x \in X$; see Figure 9.2. Using the latter with $x \in X$ would get us one vertex in $X$ "for free" as in the case of $P_2$ and $P_5$. However, it requires that all neighbors of $v_1$ and $v_3$ are endpoints of the solution, but those are never "for free" in the sense of using the local budget of any $P_3$. This can be leveraged (as we did) to control which $P_3$'s may be used to "buy" vertices in $X$; with some more work one gets a lower bound.

The kernelization for an instance of Edge Dominating Set parameterized by the size of a given modulator to a $P_5$-component graph is on the one hand easier than the kernelization of Edge Dominating Set parameterized by the size of a degree-1-modulator because we do not have connected components of size one, and on the other hand more difficult because the middle vertex of the $P_5$ does not have the property that there exists a local optimum solution that uses an edge between this middle vertex and the modulator. This leads to the following result which also follows from Lemma 10.15 and Remark 10.18 in Chapter 10. Nevertheless, we will sketch the proof to get a better understanding why we can still reduce the number of connected components that are isomorphic to the $P_5$.

**Theorem 9.4.** Edge Dominating Set *parameterized by the size of a given modulator $X$ to a $P_5$-component graph admits a kernel with $\mathcal{O}(|X|)$ vertices.*

**Proof sketch.** Let $(G, k, X)$ be an instance of Edge Dominating Set parameterized by the size of a modulator to a $P_5$-component graph, and let $\mathcal{C}$ be the set of connected components of $G - X$. We construct a bipartite graph $G_B$ where one part is the set $X$, the other part consists of one vertex $s_P$ for every connected component $P$ in $\mathcal{C}$, and where there is an edge between $x \in X$ and $s_P$ with $P = w_1w_2w_3w_4w_5 \in \mathcal{C}$ if and only if $x$ is adjacent to a vertex of $P$ that is not the middle vertex $w_3$. As in the proof of Theorem 9.2 this bipartite graph matches a connected component of $G - X$ with a vertex $x$ in $X$ if and only if we can use this component to cover this vertex $x$ "for free" using a local optimum solution that contains an edge between $x$ and this component.

Now, we apply Theorem 2.2 to obtain either a maximum matching in $G_B$ that saturates $X$ or a set $Y \subseteq X$ such that $|N_{G_B}(Y)| < |Y|$ and such that there exists a maximum matching in $G_B - N_{G_B}[Y]$ that saturates $X \setminus Y$. If there exists a maximum matching in $G_B$ that saturates $X$ then let $X_1 = X$ and $X_2 = \emptyset$. Otherwise, if there exists a set $Y$ with the above properties then let $X_1 = X \setminus Y$ and $X_2 = Y$. Observe that $X_2$ also contains the vertices in $X$ that are only adjacent to middle vertices of components in $\mathcal{C}$, and the vertices in $X$ that are not adjacent to any component in $\mathcal{C}$. Let $M$ be a maximum matching in $G_B - N_{G_B}[X_2]$ that saturates $X_1$. The partition $X_1 \dot\cup X_2$ of $X$ fulfills the following properties:

- Let $\mathcal{C}_2$ be the set of connected components $P$ in $\mathcal{C}$ where $s_P$ is a vertex in $N_{G_B}(X_2)$, i.e., $\mathcal{C}_2 = \{P = w_1w_2w_3w_4w_5 \in \mathcal{C} \mid N_G(\{w_1, w_2, w_4, w_5\}) \cap X_2 \neq \emptyset\}$. It holds either that $\mathcal{C}_2$ is the empty set (when $X_2 = \emptyset$) or that it contains less than $|X_2|$ connected components of $\mathcal{C}$, i.e., $|\mathcal{C}_2| < |X_2|$ (when $Y = X_2 \neq \emptyset$).

- For every vertex $x \in X_1$, let $P_x = w_1^x w_2^x w_3^x w_4^x w_5^x$ be the connected component in $\mathcal{C}_1 := \mathcal{C} \setminus \mathcal{C}_2$ that is paired to $x$ by $M$, i.e., $\{x, s_{P_x}\} \in M$. It holds that there exists a vertex $w^x \in \{w_1^x, w_2^x, w_4^x, w_5^x\}$ such that $\{w^x, x\} \in E(G)$ (definition of $G_B$). Note that $\mathcal{C}_1$ also contains all connected components that are not adjacent to any vertex in $X$ or where only the middle vertex of a path in $\mathcal{C}$ is adjacent to a vertex in $X$.

Using the above partition, one can show that there exists an optimum solution $S$ that contains for each path $P_x$ with $x \in X_1$ the locally optimal solution $\{\{x, w^x\}, \{w_3^x, w_2^x\}\}$ or $\{\{x, w^x\}, \{w_3^x, w_4^x\}\}$ depending on whether $w^x \in \{w_4^x, w_5^x\}$ or $w^x \in \{w_1^x, w_2^x\}$, respectively. More generally, for every vertex $w$ of a path $P \in \mathcal{C}$, except the middle vertex, and every vertex $x \in X$ that is adjacent to $w$ there exists a local optimum solution to $P$ that uses edge $\{w, x\}$ and has the middle vertex of $P$ as an endpoint of the second solution edge. As mentioned above, this is the crucial difference to a path $P' = v_1v_2v_3$ of length two. Here, the only locally optimal solution that dominates $P'$ and contains an edge between $P'$ and $X$ is the edge $\{v_2, x\}$ with $x \in X$, but this local solution does not contain the vertex $v_1$ or $v_3$. We used this in our lower bound construction to control which $P_3$'s may be used to "buy" vertices in $X$.

**Reduction Rule 9.4.** Delete $X_1$ from $G$, i.e., let $G' = G - X_1$, $X' = X \setminus X_1 = X_2$, and $k' = k$.

Claim 9.5. *Reduction Rule 9.4 is safe.*

Proof. As in the proof of Claim 9.3 we can construct an edge dominating set $F'$ of size at most $k' = k$ in $G'$ when given an edge dominating set $F$ of size at most $k$ in $G$ by replacing every edge $e = \{x, y\} \in F$ with $x \in X$, $y \notin X$ and $\delta_{G'}(y) \neq \emptyset$ by exactly one edge in $\delta_{G'}(y)$, and by deleting all other edges that have one endpoint in $X_1$.

For the other direction, let $F'$ be an edge dominating set of size at most $k'$ in $G'$. Here, we have to be a bit more careful with the middle vertices of the $P_5$'s where we change the local solution. Consider the path $P_x = w_1^x w_2^x w_3^x w_4^x w_5^x$ for some vertex $x \in X_1$. It holds that the only vertex in $P_x$ that can be adjacent to a vertex in $X' = X \setminus X_1 = X_2$ is vertex $w_3^x$; otherwise $P_x$ would be a component in $\mathcal{C}_2$ and not in $\mathcal{C}_1$ (by definition of $\mathcal{C}_1$ and $\mathcal{C}_2$). Furthermore, the edge dominating set $F'$ must dominate the two non-adjacent edges $\{w_1^x, w_2^x\}$ and $\{w_4^x, w_5^x\}$. Since $w_1^x$, $w_2^x$, $w_4^x$, and $w_5^x$ are only adjacent to vertices in $P_x$ the set $F'$ must contain one of the two edges $e_{1,2}^x = \{w_1^x, w_2^x\}$, $e_{2,3}^x = \{w_2^x, w_3^x\}$ and one of the two edges $e_{3,4}^x = \{w_3^x, w_4^x\}$, $e_{4,5}^x = \{w_4^x, w_5^x\}$. To obtain an edge dominating set of size at most $k$ in $G$ we replace for each vertex $x \in X_1$ these edges with the local optimum solution $\{\{x, w^x\}, \{w_3^x, w_2^x\}\}$ or $\{\{x, w^x\}, \{w_3^x, w_4^x\}\}$ depending whether

$w^x \in \{w_4^x, w_5^x\}$ or $w^x \in \{w_1^x, w_2^x\}$, respectively. It holds that $|F| \leq |F'|$ because for every vertex $x \in X_1$ we replace the at least two edges in $F' \cap \{e_{1,2}^x, e_{2,3}^x, e_{3,4}^x, e_{4,5}^x\}$ by the two edges of the locally optimal solution $\{\{x, w^x\}, \{w_3^x, w_2^x\}\}$ or $\{\{x, w^x\}, \{w_3^x, w_4^x\}\}$.

It remains to show that $F$ is indeed an edge dominating set in $G$. The set $V(F)$ contains all vertices in $V(F')$, except some vertices in the connected components $P_x$ with $x \in X_1$ where we change the edge dominating set $F'$. Furthermore, $V(F)$ contains all vertices in $X_1$ because for every vertex $x \in X_1$ the edge $\{w^x, x\}$ is contained in $F$. Thus, the only edges that are possibly not dominated by $F$ have one endpoint in a path $P_x$ with $x \in X_1$. Since $w_3^x$ is contained in $V(F)$ (by construction), since every edge in $P_x$ is dominated by $F$ (by construction), and since the vertices in $\{w_1^x, w_2^x, w_4^x, w_5^x\}$ are only adjacent to vertices in $P_x \cup X_1$, it follows that $F$ is an edge dominating set in $G$. □

After applying Reduction Rule 9.4 it holds that for each path $P = w_1w_2w_3w_4w_5 \in \mathcal{C}_1$ only the vertex $w_3$ can be adjacent to a vertex in $X$, and we can assume that every (optimum) solution contains the edges $\{w_2, w_3\}$ and $\{w_3, w_4\}$. Additionally, one can show that there exists an optimum solution that does not contain any edge between $\mathcal{C}_1$ and $X$ because we can replace any such edge $e = \{x, v\}$ with $v \in V(\mathcal{C}_1)$ by the edge $\{x, u\}$ with $u \in N_G(x) \setminus V(\mathcal{C}_1)$ (or delete this edge when $N_G(x) \setminus V(\mathcal{C}_1) = \emptyset$). This allows us to delete $\mathcal{C}_1$ from $G$.

Observe that this part differs from the previous kernelization where at most $|X|$ connected components (that consist of two vertices) are adjacent to vertices in $X$ after deleting $X_1$. Therefore, we could delete every connected component that is not adjacent to any vertex in $X$. This is not the case here, because of the middle vertices of the $P_5$'s. To this end, it is a bit more complicated to prove the correctness of the following reduction rule.

**Reduction Rule 9.5.** Delete all connected components in $\mathcal{C}_1$ and decrease $k$ by the size of a minimum edge dominating set in $\mathcal{C}_1$, i.e., let $G' = G - \mathcal{C}_1$, $X' = X$, and $k' = k - \textsc{eds}(\mathcal{C}_1)$.

Claim 9.6. *Reduction Rule 9.5 is safe.*

Proof. First, we will show that there exists an edge dominating set $F$ of size at most $k$ in $G$ such that no edge in $F$ has one endpoint in a connected component of $\mathcal{C}_1$ and the other endpoint in $X$. Let $F$ be an edge dominating set of size at most $k$ in $G$ with $F \cap E(\mathcal{C}_1, X)$ minimal, and let $P = w_1w_2w_3w_4w_5$ be a path in $\mathcal{C}_1$. We can assume, without loss of generality, that $F$ contains the edges $\{w_2, w_3\}$ and $\{w_3, w_4\}$ because $F$ must dominate the non-adjacent edges $\{w_1, w_2\}$, $\{w_4, w_5\}$, and the vertices $w_1$, $w_2$, $w_4$, $w_5$ are only adjacent to vertices in $P$; otherwise, $P$ is contained in $\mathcal{C}_2$ and not $\mathcal{C}_1$. Now, assume for contradiction that there exists an edge $e = \{x, y\} \in F \cap E(\mathcal{C}_1, C)$ with $x \in X$ and $y \in P$ where $P = w_1w_2w_3w_4w_5$ is a path in $\mathcal{C}_1$. It holds that $y = w_3$ because $w_3$ is the only vertex in $P$ that is adjacent to a vertex in $X$. If every vertex $u \in N_G(x)$ is contained in $V(F)$ then let $\widetilde{F} = F \setminus \{e\}$. Otherwise, let $\widetilde{F} = F \setminus \{e\} \cup \{\{x, u\}\}$, where

$u \in N_G(x) \setminus V(F)$. It holds that $\widetilde{F}$ is an edge dominating set in $G$ because $y = w_3$ is still a vertex in $V(\widetilde{F})$ which implies $V(F) \subseteq V(\widetilde{F})$. Furthermore, $u$ is not contained in a connected component of $\mathcal{C}_1$ because for every path $P = w_1w_2w_3w_4w_5$ in $\mathcal{C}_1$ the vertex $w_3$ is contained in $V(F)$ and no other vertex is adjacent to a vertex in $X$. Now, the set $\widetilde{F}$ is an edge dominating set of size at most $k$ in $G$ with $\widetilde{F} \cap E(\mathcal{C}_1, X) \subsetneq F \cap E(\mathcal{C}_1, X)$ which contradicts the minimality of $F \cap E(\mathcal{C}_1, X)$ and proves that there exists an edge dominating set $F$ of size at most $k$ in $G$ with $F \cap E(\mathcal{C}_1, X) = \emptyset$. This implies that $F' = F \setminus E(\mathcal{C}_1)$ is an edge dominating set of size at most $k'$ in $G'$ when $F$ is a solution to $(G, k, X)$ with $F \cap E(\mathcal{C}_1, X) = \emptyset$.

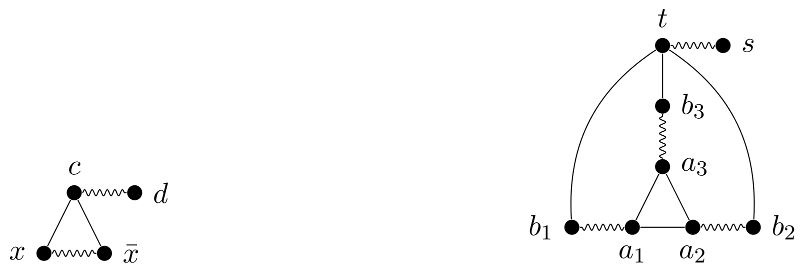
For the other direction, let $F'$ be an edge dominating set of size at most $k'$ in $G'$. To obtain an edge dominating set $F$ of size at most $k$ in $G$ we add for every path $P = w_1w_2w_3w_4w_5$ in $\mathcal{C}_1$ the two edges $\{w_2, w_3\}$ and $\{w_3, w_4\}$, which are a minimum edge dominating set of $P$, to $F'$. It follows that $F$ has size $|F'| + \text{EDS}(\mathcal{C}_1) \leq k$. The set $F$ dominates all edges in $G - X$ as well as all edges between $\mathcal{C}_2$ and $X$ because $F' \subseteq F$, and because $F$ contains an edge dominating set of $\mathcal{C}_1$. Additionally, $F$ dominates all edges between $\mathcal{C}_1$ and $X$ because $F$ dominates all middle vertices of the paths in $\mathcal{C}_1$ which are the only vertices in $\mathcal{C}_1$ that are adjacent to $X$. Hence, $F$ is an edge dominating set of size at most $k$ in $G$. □

Let $(G', k', X')$ be the reduced instance. It holds that the set of connected components in $G' - X'$ is $\mathcal{C}_2$ because we delete all other connected components during Reduction Rule 9.5. Since $|\mathcal{C}_2| \leq |X_2| = |X'|$ it follows that $G'$ has at most $5 \cdot |\mathcal{C}_2| + |X'| \leq 6|X'|$ vertices. It remains to show that we can perform the reduction in polynomial time. We apply each reduction rule at most once. Furthermore, we can apply the reduction rules in polynomial time because we can compute the partition of $X$ as well as the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ in polynomial time, and because we can delete sets of vertices from $G$ and $X$ in polynomial time. ■

While this is not the full story about the classification in the following chapter, it hopefully shows the spirit of how upper and lower bounds for kernelization can arise. Solution edges between components of $G - X$ and $X$ play a crucial role and they affect the solutions for components in nontrivial ways, e.g., apart from control opportunities, it depends on how much budget is needed for $H - B$ when edges between $B$ and $X$ are in the solution.

## 9.5. Above Half a Maximum Matching

A natural lower bound for the size of a minimum edge dominating set is $\frac{1}{2}\text{MM}$, where MM denotes the size of a maximum matching. We show that EDGE DOMINATING SET is NP-hard even for the special case where the input graph has a perfect matching and we need to determine whether there is an edge dominating set of at most half the size of that matching. This implies that EDGE DOMINATING SET parameterized by $l = k - \frac{1}{2}\text{MM}$, where $k$ is the solution size, is para-NP-hard.




(a) Gadget for a variable $x \in X$ (b) Gadget for a clause $C = \{\lambda_1, \lambda_2, \lambda_3\} \in \mathcal{C}$

Figure 9.3.: The wavy edges are the edges that are contained in the perfect matching $M$.

**Theorem 9.7.** Edge Dominating Set *parameterized by* $k - \frac{1}{2}\text{MM}$ *is para-*NP*-hard.*

**Proof.** To prove the theorem we show that it is NP-hard to decide whether a given graph, that has a perfect matching, has an edge dominating set of size equal to half the size of this matching. We will show that this problem is NP-hard by giving a reduction from 3-SAT (which is known to be NP-complete [Coo71]). Let $(X, \mathcal{C})$ be an instance of 3-SAT with $n$ variables and $m$ clauses. We construct a graph $G$ as follows.

For each variable $x \in X$ we construct a variable gadget (see Figure 9.3a for an illustration) consisting of four vertices $x$, $\bar{x}$, $c$, $d$, where the vertices $x$, $\bar{x}$, $c$ form a clique and the vertex $d$ is only adjacent to vertex $c$. Here $x$ and $\bar{x}$ are the literals of variable $x$. For every clause $C = \{\lambda_1, \lambda_2, \lambda_3\} \in \mathcal{C}$ we construct a clause gadget (see Figure 9.3b for an illustration) consisting of eight vertices $a_1$, $a_2$, $a_3$, $b_1$, $b_2$, $b_3$, $s$, and $t$. The vertices $a_1$, $a_2$, and $a_3$ form a clique, each vertex $a_i$, with $i \in [3]$, is also adjacent to $b_i$, and vertex $t$ is adjacent to $b_1$, $b_2$, $b_3$, and $s$. For every clause $C = \{\lambda_1, \lambda_2, \lambda_3\} \in \mathcal{C}$ we make vertex $\lambda_i$ (which is contained in a variable gadget) for $i \in [3]$ adjacent to vertex $a_i$ in the clause gadget $C$.

It is easy to verify that $G$ has a perfect matching, e.g. the edges $\{x, \bar{x}\}$, $\{c, d\}$ in every variable gadget together with the edges $\{a_i, b_i\}$ for $i \in [3]$, and the edge $\{s, t\}$ in every clause gadget are a perfect matching in $G$. (In Figure 9.3 the matching edges are the wavy edges.) We denote this maximum matching by $M$. Note that the matching $M$ has size $2n + 4m$; two edges in every variable gadget and four edges in every clause gadget.

We will show that $G$ has an edge dominating set of size $n + 2m$ if and only if the 3-SAT instance has a satisfying assignment. There cannot be an edge dominating set of smaller size, because every edge dominating set has at least half the size of any (maximum) matching.

Suppose first that the 3-SAT instance $(X, \mathcal{C})$ has a satisfying assignment $\phi\colon X \to \{\text{true}, \text{false}\}$. We construct an edge dominating set $F$ of $G$ by selecting edge $\{x, c\}$ in the variable gadget for $x \in X$ to $F$ if $x$ is set to true via $\phi$ and by selecting edge $\{\bar{x}, c\}$ in the variable gadget $x \in X$ to $F$ if $x$ is set to false via $\phi$. For every clause $C = \{\lambda_1, \lambda_2, \lambda_3\}$ we choose one true literal, without loss of generality say $\lambda_1$ is true, and we add the edges $\{t, b_1\}$ and $\{a_2, a_3\}$ to $F$. By construction, the set $F$ contains exactly $n + 2m$ edges. We have to show that $F$ is an edge dominating set of $G$.

Assume for contradiction that there exists an edge $e$ in $G$ that is not dominated by $F$. By construction of $F$, this edge cannot be in a clause or a variable gadget. Hence, this edge must have one endpoint in a clause gadget and one endpoint in a variable gadget. Let $C = \{\lambda_1, \lambda_2, \lambda_2\}$ be the clause that corresponds to the clause gadget that contains one endpoint of $e$. By construction of $G$ and $F$, the endpoint of $e$ is exactly the vertex $a_i$ that is not contained in $V(F)$. This implies that literal $\lambda_i$ is true. Since $\lambda_i$ is the only neighbor of $a_i$ outside the clause gadget, $\lambda_i$ is the other endpoint of $e$. But $\lambda_i$ is contained in $V(F)$ (by construction), hence $e$ is dominated.

Now, suppose $G$ has an edge dominating set $F$ of size $n + 2m$. Since the matching $M$ has twice the size of the edge dominating set $F$, it must hold that every edge in $F$ dominates two matching edges and different edges in $F$ dominate different matching edges; otherwise there would be an edge that is not dominated by $F$. The matching edge $\{c, d\}$ in a variable gadget for variable $x \in X$ has as neighbors only the vertices $x$ and $\bar{x}$, therefore either edge $\{c, x\}$ or edge $\{c, \bar{x}\}$ is contained in the edge dominating set $F$.

To satisfy the instance $(X, \mathcal{C})$ of 3-SAT let $x$ be set to true if $\{x, c\} \in F$ and $x$ be set to false if $\{\bar{x}, c\} \in F$. Exactly one of these edges is contained in $F$ (see above). To show that this is a satisfying assignment consider an arbitrary clause $C = \{\lambda_1, \lambda_2, \lambda_3\} \in \mathcal{C}$ and the clause gadget for clause $C$.

The matching edge $\{s, t\}$ in the clause gadget for clause $C$ has as neighbors only the vertices $b_1$, $b_2$, and $b_3$, hence exactly one of the edges $\{t, b_1\}$, $\{t, b_2\}$, $\{t, b_3\}$ is contained in $F$. Assume, without loss of generality, that $\{t, b_1\} \in F$, which dominates the matching edge $\{a_1, b_1\}$. Thus, no other edge of $F$ can also dominate this edge $\{a_1, b_1\}$ and, hence, $a_1 \notin V(F)$. Now, however, the edge $\{a_1, \lambda_1\}$ is only dominated by $F$ if $\lambda_1 \in V(F)$, which holds only if $\{c, \lambda_1\}$ is contained in $F$. By construction of our assignment in the previous paragraph, this implies that $\lambda_1$ is true and that clause $C$ is satisfied. This shows that $(X, \mathcal{C})$ is satisfiable and completes the proof. ■

The graph we construct in the proof of Theorem 9.7 is also a Kőnig graph, i.e., the size of a minimum vertex cover is equal to the size of a maximum matching. This implies that Edge Dominating Set restricted to Kőnig graphs is also NP-hard (even if $k = \frac{1}{2}\text{MM}$). This even rules out fpt-algorithms for the most natural above lower bound parameter $\ell = k - \frac{1}{2}\text{MM}$.

## 9.6. Summary

We showed that it is unlikely that Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph has a polynomial kernel which rules out polynomial kernels for the structural parameters modulator to a linear forest and forest. Therefore, it seems hopeless to believe that there are interesting structural parameters for which Edge Dominating Set has a polynomial kernel. However,

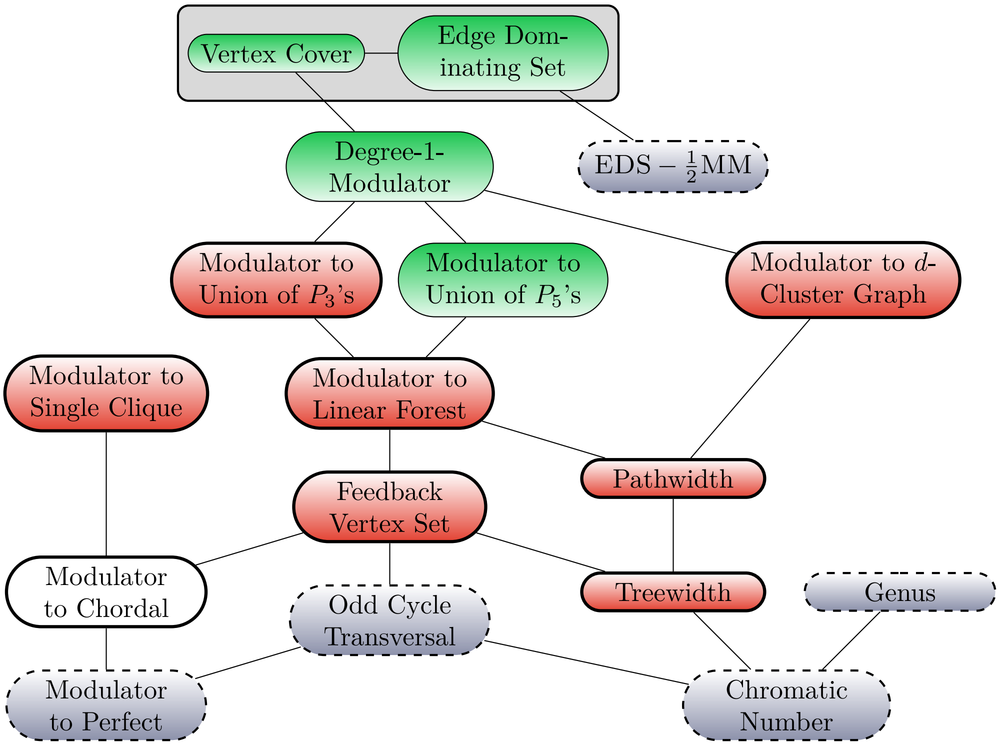



Figure 9.4.: Parameter hierarchy for Edge Dominating Set, where we assume that the modulator is given in the input. The shadings indicated that the parameterization is either para-NP-complete , FPT but conditionally lacking a polynomial kernel , FPT with polynomial kernel , or unkown whether the problem is NP-hard for constant parameter value . A line between two parameters indicates that the lower parameter can be bounded in a function of the higher parameter. The two parameters that are grouped are equivalent up to constant factors.

the kernel for $P_5$-component graphs somehow indicates that Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graphs behaves in a strange way. The polynomial kernel for Edge Dominating Set parameterized by the size of a degree-1-modulator completes the picture for Edge Dominating Set parameterized by the size of modulators for hereditary graph classes (because it is unlikely that Edge Dominating Set parameterized by the size of a modulator to a $K_3$-component graph has no polynomial kernel). Our other result, NP-hardness for Edge Dominating Set on graphs with a perfect matching of size MM, even for $k = \frac{1}{2}$MM, leaves little hope for tractability above tight lower bounds. The results are summarized in Figure 9.4, whereby we did not show the result for Edge Dominating Set parameterized by the size of a Modulator to a $d$-cluster graph so far. However, this result is covered by Theorem 10.4 item 1c.

# CHAPTER 10 CLASSIFICATION FOR $\mathcal{H}$-COMPONENT GRAPHS

## 10.1. Introduction

We have seen in the previous chapter that Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph has no polynomial kernel (unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$), and that Edge Dominating Set parameterized by the size of a modulator $X$ to a $P_5$-component graph has a polynomial kernel with $\mathcal{O}(|X|)$ vertices. This indicates that the structure even of constant-sized components permitted in $G - X$ determines in a nontrivial way whether or not there is a polynomial kernelization when parameterized by $|X|$. Note the contrast with Vertex Cover where a modulator to component size $d$ admits a kernelization with $\mathcal{O}(k^d)$ vertices for each fixed $d$. Naturally, we are interested in finding out exactly which cases admit polynomial kernels.

This brings us to our main result of this part. We fully classify the existence of polynomial kernels for parameterization by the size of a modulator to the class of $\mathcal{H}$-component graphs for all finite sets $\mathcal{H}$ of graphs. To clarify, the input consists of $(G, k, X)$ such that $G - X$ is an $\mathcal{H}$-component graph and the task is to determine whether $G$ has an edge dominating set of size at most $k$; the parameter is $|X|$. Note that these problems are fixed-parameter tractable for all finite sets $\mathcal{H}$ because $G$ has treewidth at most $|X| + \mathcal{O}(1)$.

**Theorem 10.1.** *For every finite set $\mathcal{H}$ of graphs, the* Edge Dominating Set *problem parameterized by the size of a given modulator $X$ to the class of $\mathcal{H}$-component graphs falls into one of the following two cases:*

1. *It has a kernelization algorithm that reduces to $\mathcal{O}(|X|^d)$ vertices, $\mathcal{O}(|X|^{d+1})$ edges, and size $\mathcal{O}(|X|^{d+1} \log |X|)$. Moreover, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, there is no kernelization to size $\mathcal{O}(|X|^{d-\varepsilon})$ for any $\varepsilon > 0$. Here $d = d(\mathcal{H})$ is a constant depending only on the set $\mathcal{H}$.*

2. *It has no polynomial kernelization unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

To obtain the classification one needs to understand how connected components of $G - X$ that are isomorphic to some graph $H \in \mathcal{H}$ can interact with a solution for $G$, and to derive properties of $H$ that can be leveraged for kernels or lower bounds for kernelization. Crucially, edge dominating sets for $G$ may contain edges between $X$ and components of $G - X$. From the perspective of such a component (isomorphic to $H$) this is equivalent to first covering edges incident with some vertex set $B \subseteq V(H)$ (the endpoints of chosen edges to $X$) and then covering the remaining edges by a minimum edge dominating set for $H - B$. Depending on the size of a minimum edge dominating set of $H - B$ and further properties of $H$, such a set $B$ may be used to rule out any polynomial kernels or to give a lower bound of $\mathcal{O}(|X|^{d-\varepsilon})$ for the kernel size, where $d = |B|$. Conversely, absence of such sets or an upper bound for their size can be leveraged for kernels. Some sets $B$ may make others redundant, complicating both upper and lower bounds.

For a given finite set $\mathcal{H}$ of graphs, the lower bound obtained from the classification is simply the strongest one over all $H \in \mathcal{H}$. If this does not already rule out a polynomial kernelization then, for each $H \in \mathcal{H}$, we can reduce the number of components isomorphic to $H$ to $\mathcal{O}(|X|^{d(H)})$ where $d(H)$ depends only on $H$. Moreover, we also prove that there is an almost matching lower bound of $\mathcal{O}(|X|^{d(H)-\varepsilon})$, assuming $\mathsf{NP} \nsubseteq \mathsf{coNP/poly}$. The value $d(\mathcal{H})$ is the maximum over all $d(H)$ for $H \in \mathcal{H}$ that yield such a polynomial lower bound; it can be computed in time depending only on $\mathcal{H}$, i.e., in constant time for each fixed set $\mathcal{H}$.

## 10.2. Modulator to an $\mathcal{H}$-Component Graph

In contrast to Vertex Cover, where we can delete a vertex in the modulator if we know that this vertex must be in a solution of certain size, this is not the case for Edge Dominating Set because we do not necessarily know which incident edge should be chosen. Of course, we can check for a vertex $x$ in the modulator $X$ how not having this vertex as an endpoint of a solution edge influences the size of a minimum edge dominating set of $G - X$. But, even if we find out that a vertex $x$ in the modulator $X$ must be an endpoint of a solution edge, we do not know if the other endpoint of the solution edge incident with $x$ is in $X$ or in a connected component of $G - X$. If there would be a connected component $C$ in $G - X$ with the property that there exists a vertex $v \in N(x) \cap V(C)$ with $\textsc{eds}(C) = \textsc{eds}(C - v) + 1$, then it could be possible to have $x$ as an endpoint of a solution edge without paying more than the cost of a minimum edge dominating set in $C$. Thus, instead of finding vertices in the modulator that must be endpoint of a solution edge, we want to find vertices in the modulator that can be endpoints of a solution edge without spending more budget than the size of a minimum edge dominating set in $G - X$. Similarly, getting edges to $r$ vertices in $X$ while increasing the cost in $C$ by less than $r$ is of interest (we can always have cost equal to $r$). The following definition classifies relevant vertices and vertex sets in a graph $H$, which may occur as a component of $G - X$.

**Definition 10.2.** Let $H = (V, E)$ be a connected graph.

- We call a vertex $v \in V$ *extendable* if $\textsc{eds}(H - v) + 1 = \textsc{eds}(H)$. We denote the set of extendable vertices of $H$ by $Q(H)$.

  *Intuitively, these vertices allow a local solution for an $H$-component in $G - X$ that includes an edge $\{v, x\}$ with $x \in X$ and $v \in V(H)$.*

- We call a set $Y \subseteq Q(H)$ *free* if for all vertices $v \in Y$ and for all minimum edge dominating sets $F$ in $H$ there exists a minimum edge dominating set $F'$ in $H - v$ of size $|F| - 1$ and with $V(F) \setminus Y \subseteq V(F')$. By $W(H)$ we denote the unique maximum free set of $H$. We call a vertex $w \in W(H)$ *free.*[1]

  *Intuitively, vertices in $Y$ can be used for solution edges between components and $X$, while covering the same vertices of $H - Y$ as any local optimum solution; thus, they cannot be used for lower bounds like for $P_3$-components.*

- We call a vertex $v \in V$ *uncovered* if no minimum edge dominating set $F$ of $H$ contains an edge incident with $v$, i.e. $v \notin V(F)$. We denote the set of uncovered vertices by $U(H)$.

  *Intuitively, $H$-components with any $v \in U(H)$ adjacent to $x \in X$ are easy to handle because $x \notin V(F)$ would imply that the local cost for $H$ increases above $\textsc{eds}(H)$.*

- For any $Y \subseteq V$ define $\operatorname{cost}(Y) := |Y| + \textsc{eds}(H - Y) - \textsc{eds}(H)$.

  *Intuitively, $\operatorname{cost}(Y)$ is equal to the additional budget that is needed for an $H$-component of $G - X$ when exactly the vertices in $Y$ have solution edges to $X$. Note that $\operatorname{cost}(\{v\}) = 0$ for all extendable vertices $v$.*

- We call a set $B \subseteq V \setminus W(H)$ *beneficial* if for all $\widetilde{B} \subsetneq B$ we have $|B| - \operatorname{cost}(B) > |\widetilde{B}| - \operatorname{cost}(\widetilde{B})$ or, equivalently, $\textsc{eds}(H - B) < \textsc{eds}(H - \widetilde{B})$. Note that this must also hold for $\widetilde{B} = \emptyset$ which implies that for all beneficial sets we have $|B| - \operatorname{cost}(B) > 0$ or, equivalently, $\textsc{eds}(H - B) < \textsc{eds}(H)$.

  *Intuitively, the solution may include $|B|$ edges between the set $B$ and some set $X' \subseteq X$ while increasing the cost for the $H$-component by exactly $\operatorname{cost}(B)$. This saves $|B| - \operatorname{cost}(B) > 0$ over taking any $|B|$ edges incident with $X'$. The condition for all $\widetilde{B} \subsetneq B$ ensures that the savings of getting $|B|$ edges at cost $\operatorname{cost}(B)$ is greater than for any proper subset.*

- We call a beneficial set $B$ *strongly beneficial* if $\operatorname{cost}(B) < \sum_{i=1}^{h} \operatorname{cost}(B_i)$ holds for all covers $B_1, B_2, \ldots, B_h \subsetneq B$ of $B$.

  *Intuitively, for a strongly beneficial set $B$ we cannot get the same number of edges to $X$ by using sets $B_i$ in several different $H$-components.*

[1] We show in Proposition 10.6 item (1) that $W(H)$ is unique.

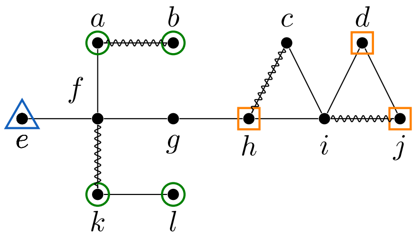



Figure 10.1.: Example of an $H$-component with $\textsc{eds}(H) = 4$. The wavy edges are a possible minimum edge dominating set of $H$.

Example 10.3 (Illustration of Definition 10.2). Figure 10.1 shows a connected graph $H$. The size of an edge dominating set in $H$ is at least four because a solution has to dominate the four pairwise non-adjacent edges $\{a, b\}, \{k, l\}, \{j, d\}$ and $\{g, h\}$. Thus, $\textsc{eds}(H) = 4$ because the wavy edges are an edge dominating set of $H$.

The vertices $\{a, b, k, l\}$, marked with a green cycle, as well as the vertices $\{d, h, j\}$, marked with an orange rectangle, are extendable. But only the green marked vertices $\{a, b, k, l\}$ are free: Let $F$ be any minimum edge dominating set in $H$. The set $F$ must contain exactly one of the two edges $e_1 = \{a, b\}$ and $e_2 = \{a, f\}$, and exactly one of the two edges $e_3 = \{k, l\}$ and $e_4 = \{k, f\}$. Now, $F' = F \setminus \{e_1, e_2, e_3, e_4\} \cup \{e_4\}$ is an edge dominating set in $H - a$ and $H - b$ of size $|F| - 1$, and $F' = F \setminus \{e_1, e_2, e_3, e_4\} \cup \{e_2\}$ is an edge dominating set in $H - k$ and $H - l$ of size $|F| - 1$ which implies that the vertices $\{a, b, k, l\}$ are free. The vertices $\{d, h, j\}$ are not free because no minimum edge dominating set $F'$ in $H - d$, respectively $H - h$, respectively $H - j$ has vertex $c$, which is not extendable, as an endpoint of a solution edge, but the graph $H$ has a minimum edge dominating set that has $c$ as an endpoint, namely the one containing the wavy edge $\{a, b\}, \{h, c\}, \{d, j\}$. The vertex $e$, marked with a blue triangle, is uncovered.

The set $\{c, g\}$ is strongly beneficial, whereas the set $\{c, g, i, j\}$ is only beneficial, but not strongly beneficial: The set $\{c, g\}$ is beneficial because $\textsc{eds}(H - \{c, g\}) = 3$ and $\textsc{eds}(H - c) = \textsc{eds}(H - g) = \textsc{eds}(H) = 4$, and strongly beneficial because the only possible non-trivial cover of $\{c, g\}$ is $\{c\}, \{g\}$ and $\text{cost}(\{c, g\}) = 1 < 2 = \text{cost}(\{c\}) + \text{cost}(\{g\})$. The set $\{c, g, i, j\}$ is beneficial because $\textsc{eds}(H - \{c, g, i, j\}) = 2$ and $\textsc{eds}(H - B) \geq 3$ for all sets $B \subsetneq \{c, g, i, j\}$. But, the set $\{c, g, i, j\}$ is not strongly beneficial because $\text{cost}(\{c, g, i, j\}) = 2 = 1 + 1 + 0 = \text{cost}(\{c, g\}) + \text{cost}(\{i\}) + \text{cost}(\{j\})$. Observe that the set $\{c, g, i\}$ is not beneficial even though $\textsc{eds}(H - \{c, g, i\}) = 3 < 4 = \textsc{eds}(H)$, because $\{c, g\} \subsetneq \{c, g, i\}$ and $\textsc{eds}(H - \{c, g, i\}) = 3 = \textsc{eds}(H - \{c, g\})$.

We are now able to give a more detailed version of Theorem 10.1, which specifies for each finite set $\mathcal{H}$ of connected graphs the kernelization complexity of Edge Dominating Set parameterized by the size of a modulator to $\mathcal{H}$-component graphs.

**Theorem 10.4.** *Let $\mathcal{H}$ be any finite set of connected graphs. The* Edge Dominating Set *problem parameterized by the size of a given modulator $X$ to an $\mathcal{H}$-component graph behaves as follows:*

1. *If $\mathcal{H}$ contains any graph $H$ fulfilling one of the following items then there is no polynomial kernelization unless* NP $\subseteq$ coNP/poly*:*
   a) *There is an extendable vertex in $H$ that is not free, i.e., $Q(H) \setminus W(H) \neq \emptyset$.*
   b) *There is a strongly beneficial set $B$ in $H$ that contains an uncovered vertex, i.e., $B \cap U(H) \neq \emptyset$.*
   c) *There is a vertex in $H$ that is neither uncovered, free, nor neighbor of a free vertex, i.e., $V(H) \setminus (N[W(H)] \cup U(H)) \neq \emptyset$.*
   d) *There is a strongly beneficial set $B \subseteq N(W(H))$ in $H$ such that no minimum edge dominating set $F_B$ of $H - B$ covers all vertices of $N(W(H)) \setminus B$.*
2. *Else, if $\mathcal{H}$ contains at least one graph that has a strongly beneficial set, then there is a kernelization algorithm that reduces to $\mathcal{O}(|X|^d)$ vertices, $\mathcal{O}(|X|^{d+1})$ edges, and size $\mathcal{O}(|X|^{d+1} \log |X|)$, and there is no kernelization algorithm that reduces to size $\mathcal{O}(|X|^{d-\varepsilon})$, for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly *where $d$ is the size of the largest strongly beneficial set in any $H \in \mathcal{H}$.*
3. *Else, there is a kernelization algorithm that reduces to $\mathcal{O}(|X|^2)$ vertices, $\mathcal{O}(|X|^3)$ edges, and size $\mathcal{O}(|X|^3 \log |X|)$, and there is no kernelization algorithm that reduces to size $\mathcal{O}(|X|^{2-\varepsilon})$, for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly.

In the remaining part of this Chapter is devoted to proving Theorem 10.4, following the proof outline below. From this, Theorem 10.1 directly follows because disconnected graphs in $\mathcal{H}$ do not affect the resulting class of $\mathcal{H}$-component graphs, i.e., given *any* finite set $\mathcal{H}$ of graphs we can take the subset $\mathcal{H}'$ of connected graphs in $\mathcal{H}$ and apply Theorem 10.4 to $\mathcal{H}'$.

Remark 10.5. We showed that Edge Dominating Set parameterized by the size of a given modulator $X$ to a $P_5$-component graph admits a kernel with $\mathcal{O}(|X|)$ vertices (see Theorem 9.4). The reason why the kernelization procedure of Theorem 10.4 item 3 only reduces to $\mathcal{O}(|X|^2)$ vertices instead of $\mathcal{O}(|X|)$ vertices is that $H$-components can have uncovered vertices. This leads to a different marking argument similar to the case for Edge Dominating Set parameterized by the solution size or the size of a degree-1-modulator. Note that Edge Dominating Set parameterized by solution size is covered by Theorem 10.4 item 3.

**Proof outline for Theorem 10.4.** We begin by establishing a number of useful properties of the terms introduced in Definition 10.2, e.g., that each graph $H$ containing a beneficial set $B$ also contains a strongly beneficial set $B' \subseteq B$ (Proposition 10.6)

The kernelization lower bound of item 1 is proved by generalizing the lower bound obtained for $P_3$-component graphs in Theorem 9.1. We define so-called *control pairs* by abstracting properties of $P_3$-components used in the proof (Definition 10.7) and show that there is no polynomial kernelization when any graph $H \in \mathcal{H}$ has a control pair (Theorem 10.8). We then show that graphs $H$ fulfilling items 1a, 1b, 1c, or 1d have control pairs (Lemmas 10.10, 10.11, 10.12, and 10.13).

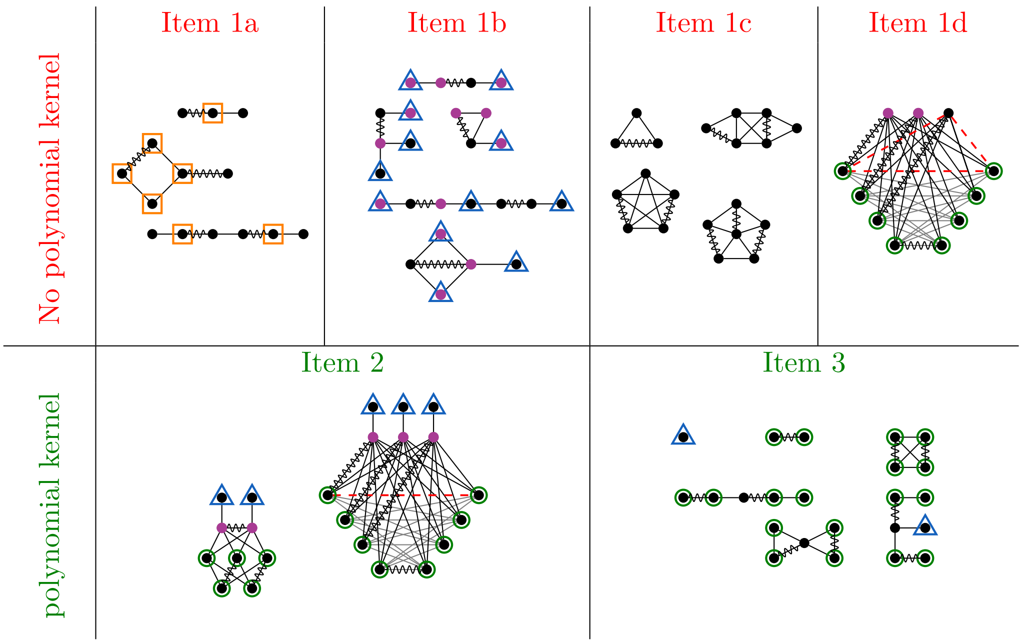



Table 10.1.: Example for the classification of Theorem 10.4. The red dashed edges are the edges that are missing to obtain locally a complete bipartite graph. The vertices that are marked with a green cycle are free, the vertices that are marked with an orange rectangle are extendable, but not free, the vertices that are marked with a blue triangle are uncovered, and the set of purple marked vertices is strongly beneficial.

In item 1d, and in the items below, we (may) use that no graph in $\mathcal{H}$ fulfills items 1a, 1b, or 1c. Accordingly, each graph $H \in \mathcal{H}$ has $V(H) = N[W(H)] \cup U(H)$, i.e., each vertex of $H$ is uncovered, free, or neighbor of a free vertex. Moreover, every extendable vertex is also free, i.e., $Q(H) = W(H)$, and strongly beneficial sets contain no (uncovered) vertices of $U(H)$. This implies that all strongly beneficial sets are subsets of $N(W(H))$, the neighborhood of the free vertices, as neither uncovered nor free vertices can be contained and no further vertices except those in $N(W(H))$ exist in $H$ (in this case).

For item 2 we have that no graph in $\mathcal{H}$ fulfills any of the items 1a through 1d and that at least one graph in $\mathcal{H}$ has a strongly beneficial set. Thus, in addition to the above restrictions on $H \in \mathcal{H}$, we know that for each strongly beneficial set $B$, which here must be a subset of $N(W(H))$, there is a minimum edge dominating set $F_B$ of $H - B$ that covers all vertices in $N(W(H)) \setminus B$. We give a general kernelization procedure that reduces the number of components in $G - X$ to $\mathcal{O}(|X|^d)$ where $d$ is the size of the largest strongly beneficial set among graphs $H \in \mathcal{H}$ (Lemma 10.19). We then rule out kernels of size $\mathcal{O}(|X|^{d-\varepsilon})$ using only $H$-components, where $H$ is any graph in $\mathcal{H}$ that exhibits the largest size $d$ of strongly beneficial sets (Lemma 10.23). Note that in the present item $d$ is always at least two because having a strongly beneficial set $B$ of size one would mean that $v \in B$ is an extendable vertex that is not free (because beneficial sets are disjoint from the set $W(H)$ of free vertices), which is handled by item 1a.

Finally, for item 3, it remains to consider the case that no graph $H \in \mathcal{H}$ fulfills any of the items 1a through 1d and that no graph in $H$ has a strongly beneficial set. It follows that no graph in $H$ has any beneficial sets (Proposition 10.6 (11)) and, as before, we have $V(H) = N[W(H)] \cup U(H)$. We obtain a kernelization to $\mathcal{O}(|X|^2)$ vertices, $\mathcal{O}(|X|^3)$ edges, and size $\mathcal{O}(|X|^3 \log |X|)$ (Lemma 10.15). The lower bound ruling out kernels of size $\mathcal{O}(|X|^{2-\varepsilon})$ for any $\varepsilon > 0$, and in fact for any set $\mathcal{H}$, follows easily by a simple reduction from Vertex Cover for which a lower bound ruling out size $\mathcal{O}(n^{2-\varepsilon})$ is known [DvM14] (Lemma 10.27).

Since the arguments required for the kernelization in item 3 are simpler than for that of item 2 and can serve as an introduction to it, the proofs are given in the order of item 1, item 3, followed by item 2. ∎

Before starting on the lower bound part of Theorem 10.4, we establish a few basic properties of the terms defined in Definition 10.2; these mostly follow readily from their definition. We also justify the definition of $W(H)$ as the unique maximum cardinality free set in $H$.

**Proposition 10.6 ([2]).** *Let $H = (V, E)$ be a connected graph, let $W = W(H)$ be the set of free vertices, let $Q = Q(H)$ be the set of extendable vertices, and let $U = U(H)$ be the set of uncovered vertices.*

1. *The set $W$ is well defined.*
2. *The set $U$ is an independent set and no vertex in $Q$ is adjacent to a vertex in $U$, i.e., $N_H(U) \cap (Q \cup U) = \emptyset$.*
3. *If $v \in N_H(U)$ is a vertex that is adjacent to a vertex in $U$, then $v$ is an endpoint of an edge in every minimum edge dominating set of $H$.*
4. *It holds for all vertices $v \in V$ that $\textsc{eds}(H) - 1 \leq \textsc{eds}(H - v) \leq \textsc{eds}(H)$.*
5. *Let $Y \subseteq V$. It holds for all subsets $X \subseteq Y$ that $\textsc{eds}(H - X) - |Y \setminus X| \leq \textsc{eds}(H - Y) \leq \textsc{eds}(H - X)$, and that $\mathrm{cost}(X) \leq \mathrm{cost}(Y)$.*
6. *Let $F$ be a minimum edge dominating set in $H$. There exists a minimum edge dominating set $F'$ in $H$ with $(V(F) \cup N_H(W)) \setminus W \subseteq V(F')$.*
7. *Every set that consists of a single vertex $v \in Q \setminus W$ is strongly beneficial. Furthermore, these are the only beneficial sets of size one.*
8. *If $B$ is a beneficial set of size at least two then $B$ contains no extendable vertex, i.e., $B \cap Q = \emptyset$.*

[2]The proof of Proposition 10.6 is deferred to Section 10.5.

9. *If there exists a set $Y \subseteq V \setminus W$ with* $\textsc{eds}(H - Y) < \textsc{eds}(H)$*, then there exists a beneficial set $B \subseteq Y$ with* $\textsc{eds}(H - B) = \textsc{eds}(H - Y)$*.*
10. *If there exists a set $Y \subseteq V \setminus W$ with* $\textsc{eds}(H - Y) < \textsc{eds}(H)$*, then there exists a beneficial set $B \subseteq Y$ with* $\textsc{eds}(H - B) + 1 = \textsc{eds}(H)$*. Furthermore, $B$ is strongly beneficial.*
11. *If $H$ has a beneficial set $B$, then $H$ has also a strongly beneficial set $B' \subseteq B$.*
12. *Let $F$ be a minimum edge dominating set in $H$. If $e = \{x, y\}$ is an edge in $F$ with $x, y \notin Q$, then $\{x, y\}$ is a strongly beneficial set.*
13. *Let $B$ be a beneficial set. $B$ is strongly beneficial if and only if for every non-trivial partition $B_1, B_2, \ldots, B_h$ of $B$ it holds that* $\operatorname{cost}(B) < \sum_{i=1}^{h} \operatorname{cost}(B_i)$*.*
14. *Let $Y \subseteq V \setminus W$. There exists a partition $B_1, B_2, \ldots, B_h$ of $Y$ where $B_i$ is either strongly beneficial or where $B_i$ has* $\operatorname{cost}(B_i) = |B_i|$*, for all $i \in [h]$, such that* $\operatorname{cost}(Y) \geq \sum_{i=1}^{h} \operatorname{cost}(B_i)$*. (Note that we also allow trivial partitions.)*

## 10.3. Generalizing the Lower Bound

We want to generalize Theorem 9.1 to get a lower bound that covers a variety of different $H$-components. In the proof of Theorem 9.1, we used one endpoint of each $P_3$ to control that we choose the edges only from one instance and to make sure that the $\binom{k}{2}$ edges have their endpoints in a set of size $k$. The middle vertex of each $P_3$ is extendable (but not free, so $P_3$ fits item 1a of Theorem 10.4) and the set consisting of this single vertex is beneficial. Hence, we were able to add edges between the middle vertices of the $P_3$'s and the set $S$ without spending more budget. Accordingly, to generalize Theorem 9.1, we define what we call control pairs consisting of a set of control vertices and a strongly beneficial set.

**Definition 10.7.** Let $H = (V, E)$ be a connected graph, let $C \subseteq V \setminus (Q(H) \cup B)$, and let $B \subseteq V$. We call the pair $(C, B)$ *control pair*, if

- $B$ is strongly beneficial,
- no vertex $c \in C$ is extendable in $H - B$, i.e., $C \cap Q(H - B) = \emptyset$,
- there exists a minimum edge dominating set $F$ in $H$ such that $C \subseteq V(F)$, and
- for all minimum edge dominating sets $F_B$ in $H - B$ it holds that $C \nsubseteq V(F_B)$.

Let $H$ be a connected graph that contains a control pair. We show that Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ for all $\mathcal{H} \ni H$. The lower bound construction generalizes the construction used for Theorem 9.1, making the proof more complicated. Observe that for $H = P_3 = v_1 v_2 v_3$ the set $B$ is the vertex $v_2$ and the set $C$ is the vertex $v_1$ (or $v_3$).

**Theorem 10.8.** *Let $H$ be a connected graph and let $B \subseteq V$, $C \subseteq V \setminus (Q(H) \cup B)$ such that $(C, B)$ is a control pair. For all sets $\mathcal{H} \ni H$ of graphs, the* Edge Dominating Set *problem parameterized by the size of a modulator to an $\mathcal{H}$-component graph admits no polynomial kernelization unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

**Proof.** We give a cross-composition from Multicolored-Clique; the theorem then follows directly from Theorem 2.14. In $G - X$ we use only components isomorphic to $H$, so $X$ is a modulator to $\mathcal{H}$-component graphs for all $\mathcal{H}$ with $H \in \mathcal{H}$.

We choose the same polynomial equivalence relation $\mathcal{R}$ as in the proof of Theorem 9.1, i.e., two instances $(G_1, k_1)$ and $(G_2, k_2)$ of Multicolored Clique are in the same equivalence class of $\mathcal{R}$ if $k_1 = k_2$ and $|V(G_1)| = |V(G_2)|$. Assume that we are given a sequence $I_i = (G_i, k)_{i=1}^t$ of Multicolored Clique instance that are in the same equivalence class of $\mathcal{R}$. Since all color classes have the same size we identify the vertex sets of each color class. Let $V$ be the vertex set (of size $k \cdot n$) of the $t$ instances and let $V_1, V_2, \ldots, V_k$ be the different color classes (each of size $n$). We assume, without loss of generality, that every instance has at least one edge in the edge set $E(V_p, V_q)$ for all $1 \leq p < q \leq k$. Otherwise, this instance would be a trivial no instance and we can delete it. Furthermore, we can assume, without loss of generality, that $t = 2^s$.

We construct an instance $(G', k', X')$ of Edge Dominating Set parameterized by the size of a modulator to an $H$-component graph. Thus, $X'$ is a modulator to $\mathcal{H}$-component graphs for all $\mathcal{H}$ with $H \in \mathcal{H}$ (see Figure 10.2 for an illustration). As in the proof of Theorem 9.1 we add sets $V$, $T = \{t_1, t_2, \ldots, t_k\}$, $T' = \{t'_1, t'_2, \ldots, t'_k\}$, $W$, $Z$, and $Z'$ to $G'$ and connect them in the same way. Again, for each instance $G_i$, with $i \in [t]$, we pick a different subset of size $s$ of $W$ and denote it by $W(i)$.

Instead of adding one vertex for each edge set $E(V_p, V_q)$ with $1 \leq p < q \leq k$ to $G'$, we add a set $S_{p,q}$ of size $d := |B|$ to $G'$ as well as a copy $S'_{p,q}$. Let $S_{p,q} = \{s^1_{p,q}, s^2_{p,q}, \ldots, s^d_{p,q}\}$, let $S'_{p,q} = \{s'^1_{p,q}, s'^2_{p,q}, \ldots, s'^d_{p,q}\}$, let $S = \bigcup_{1 \leq p < q \leq k} S_{p,q}$, and let $S' = \bigcup_{1 \leq p < q \leq k} S'_{p,q}$. We make every vertex $s^j_{p,q}$ with $1 \leq p < q \leq k$ and $j \in [d]$ adjacent to vertex $s'^j_{p,q}$. For each graph $G_i$, with $i \in [t]$, we add $|E(G_i)|$ copies of graph $H$ to $G'$. We denote by $H^e_i$ the copy of $H$ that represents edge $e = \{x, y\} \in E(G_i)$ of instance $i \in [t]$ and by $(C^e_i, B^e_i)$ the control pair of $H^e_i$. For edge $e = \{x, y\}$ in instance $G_i$, we make every vertex in $C^e_i$ adjacent to all vertices in $W(i)$ and to the vertices $x, y \in V$.

To be able to refer to single vertices of $B$ in copies $H^e_i$ of $H$, let $B = \{b_1, \ldots, b_d\}$ and let $B^e_i = \{b^e_{i,1}, b^e_{i,2}, \ldots, b^e_{i,d}\}$ where $b^e_{i,j}$ corresponds to $b_j$ in $B$ (i.e., this correspondence constitutes an isomorphism between $H$ and $H^e_i$). For all $i \in [t]$, $e \in E(G_i)$, and $j \in [d]$ we make vertex $b^e_{i,j} \in B^e_i$ adjacent to vertex $s^j_{p,q} \in S_{p,q}$ if $e \in E(V_p, V_q)$ for $1 \leq p < q \leq k$. Note that every vertex $b^e_{i,j}$ is adjacent to exactly one vertex in $S$.

The modulator $X'$ contains all vertices that are not contained in a copy of $H$. Thus $X' = V(G') \setminus \bigcup_{i=1}^t \bigcup_{e \in E(G_i)} V(H^e_i) = V \cup T \cup T' \cup W \cup Z \cup Z' \cup S \cup S'$ and $X'$ has size $|X'| = k \cdot n + 2k + 4s + 2 \cdot \binom{k}{2}|B|$. Observe that the size of $X'$ is polynomially bounded in $n + s$, because $k \leq n$ and $|B|$ is constant. Let $k' = s + k + \sum_{i=1}^t |E(G_i)| \cdot \textsc{eds}(H) + \binom{k}{2} \cdot \operatorname{cost}(B)$. Note that $0 \leq \operatorname{cost}(B) < |B|$, because $B$ is a strongly beneficial set in $H$.

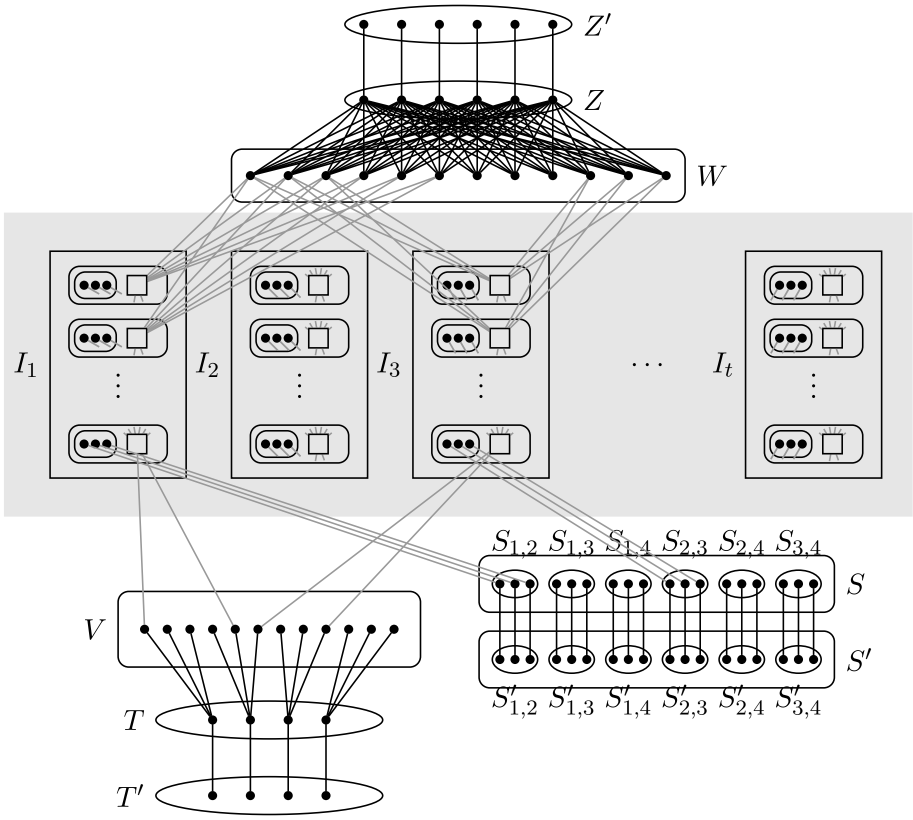



Figure 10.2.: Construction of the graph $G'$ with $k = 4$ and $d = 3$, where $X' = W \cup Z \cup Z' \cup V \cup T \cup T' \cup S \cup S'$. For each graph $H$, the rectangle corresponds to the set $C$ and the ellipse to the strongly beneficial set $B$.

We will show that $(G', k', X')$ is a yes-instance of Edge Dominating Set if and only if there is an $i^* \in [t]$ such that $(G_{i^*}, k)$ is a yes-instance of Multicolored Clique.

($\Rightarrow$:) Let $F$ be an edge dominating set of size at most $k'$ in $G'$. Analogously to the proof of Theorem 9.1 we can observe, that $Z \cup T \cup S \subseteq V(F)$ and that we can choose the $|T|$, $|Z|$ and $|S|$ edges that have one endpoint in $T$, $Z$ or $S$, respectively, always from the edge set $E(T, V)$, $E(Z, W)$ or $E(S, \bigcup_{i=1}^{t} \bigcup_{e \in E(G_i)} H_i^e)$, respectively. Additionally, we can assume, without loss of generality, that the edges in $F \cap E(T, V)$, $F \cap E(Z, W)$ and $F \cap E(S, \bigcup_{i=1}^{t} \bigcup_{e \in E(G_i)} H_i^e)$ are a matching (simple replacement argument). Furthermore, an edge in $F$ cannot have its endpoints in different copies of $H$.

Let $F_T = F \cap E(T, V)$, let $F_Z = F \cap E(Z, W)$, let $F_S = F \cap E(S, \bigcup_{i=1}^{t} \bigcup_{e \in E(G_i)} H_i^e)$, and let $F_R = F \setminus (F_T \cup F_Z \cup F_S)$. Recall, the edge sets $F_T$, $F_Z$ and $F_S$ are matchings in $G'$. We can assume that every edge in $F_R$ has at least one endpoint in $V(G') \setminus X' = \bigcup_{i=1}^{t} \bigcup_{e \in E(G_i)} V(H_i^e)$, because every edge in $E(G'[X'])$ is dominated by an edge in $F_T \cup F_Z \cup F_S$. This follows from the fact that $Z \cup T \cup S$ are covered by the edge dominating set $F$ and that $X' \setminus (Z \cup T \cup S)$ is an independent set.

Let $B_i^e(F) = \{v \in B_i^e \mid \exists s \in S\colon \{v,s\} \in F_S\}$ be the set of vertices in $B_i^e$ that are incident with an edge in $F_S$, where $i \in [t]$ and $e \in E(G_i)$. Since $F_S$ is a matching between $S$ and $\bigcup_{i=1}^t \bigcup_{e\in E(G_i)} H_i^e$ which covers $S$ and contains all edges of $F$ that are incident with $S$, and since $|F_S| = |S|$ it holds that no edge in $F \setminus F_S$ has an endpoint in $S$ and that $\sum_{i=1}^t \sum_{e\in E(G_i)} |B_i^e(F)| = |S| = \binom{k}{2} \cdot |B|$. For $i \in [t]$ and $e \in E(G_i)$, let $F_i^e = \{f \in F_R \mid f \cap V(H_i^e) \neq \emptyset\}$ be the set of edges in $F_R$ that have at least one endpoint in $H_i^e$. It holds that $\bigcup_{i=1}^t \dot\bigcup_{e\in E(G_i)} F_i^e = F_R$, because every edge in $F_R$ is incident with a vertex in $\bigcup_{i=1}^t \bigcup_{e\in E(G_i)} V(H_i^e)$. Moreover, the sets $F_i^e$ are a partition of $F_R$ because no edge is incident with vertices of different graphs $H_i^e$.

Every edge in the graph $H_i^e$, with $i \in [t]$ and $e \in E(G_i)$, must be dominated by edges in $F_R$ and $F_S$, because they cannot be dominated by edges in the vertex set $F_T \cup F_Z \subseteq E(T,V) \cup E(Z,W)$. Thus, $F_i^e$ must dominate all edges in $H_i^e - B_i^e(F)$, because the set $B_i^e(F)$ contains all vertices in $H_i^e$ that are incident with edges in $F_S$. This implies that for all $i \in [t]$ and $e \in E(G_i)$, the set $F_i^e$ has at least the size of a minimum edge dominating set in $H_i^e - B_i^e(F)$; hence $\textsc{eds}(H_i^e - B_i^e(F)) \leq |F_i^e|$. Combining all this, we get:

$$
\begin{aligned}
|F| &= |F_T| + |F_Z| + |F_S| + |F_R| \\
&= k + s + \binom{k}{2}|B| + \sum_{i=1}^{t} \sum_{e\in E(G_i)} |F_i^e| \\
&\qquad \text{because the sets } F_i^e \text{ partition } F_R \\
&\geq k + s + \binom{k}{2}|B| + \sum_{i=1}^{t} \sum_{e\in E(G_i)} \textsc{eds}(H_i^e - B_i^e(F)) \\
&\qquad \text{because } \textsc{eds}(H_i^e - B_i^e(F)) \leq |F_i^e| \\
&= k + s + \binom{k}{2}|B| + \sum_{i=1}^{t} \sum_{e\in E(G_i)} (\textsc{eds}(H_i^e) - |B_i^e(F)| + \operatorname{cost}(B_i^e(F))) \\
&\qquad \text{by the definition of } \operatorname{cost}(B_i^e(F)) \\
&= k + s + \binom{k}{2}|B| + \sum_{i=1}^{t} |E(G_i)|\,\textsc{eds}(H) - \binom{k}{2}|B| + \sum_{i=1}^{t} \sum_{e\in E(G_i)} \operatorname{cost}(B_i^e(F)) \\
&\qquad \text{because } \sum_{i=1}^{t} \sum_{e\in E(G_i)} |B_i^e(F)| = |S| = \binom{k}{2} \cdot |B| \\
&= k + s + \sum_{i=1}^{t} |E(G_i)|\,\textsc{eds}(H) + \sum_{i=1}^{t} \sum_{e\in E(G_i)} \operatorname{cost}(B_i^e(F)) \\
&= k + s + \sum_{i=1}^{t} |E(G_i)|\,\textsc{eds}(H) + \sum_{1\leq p<q\leq k} \sum_{\substack{x\in V_p \\ y\in V_q}} \sum_{\substack{i\in[t] \\ \{x,y\}\in E(G_i)}} \operatorname{cost}(B_i^{\{x,y\}}(F))
\end{aligned}
$$

Because we also have that

$$|F| \leq k' = k + s + \sum_{i=1}^{t} |E(G_i)| \operatorname{EDS}(H) + \binom{k}{2} \operatorname{cost}(B),$$

it follows directly that

$$\sum_{1 \leq p < q \leq k} \sum_{\substack{x \in V_p \\ y \in V_q}} \sum_{\substack{i \in [t] \\ \{x,y\} \in E(G_i)}} \operatorname{cost}(B_i^{\{x,y\}}(F)) \leq \binom{k}{2} \cdot \operatorname{cost}(B). \tag{10.1}$$

Therefore, there exist $1 \leq \bar{p} < \bar{q} \leq k$ such that

$$\sum_{\substack{x \in V_{\bar{p}} \\ y \in V_{\bar{q}}}} \sum_{\substack{i \in [t] \\ \{x,y\} \in E(G_i)}} \operatorname{cost}(B_i^{\{x,y\}}(F)) \leq \operatorname{cost}(B).$$

Claim 10.9. *The set* $B_C := \bigcup_{\substack{x \in V_{\bar{p}} \\ y \in V_{\bar{q}}}} \dot{\bigcup}_{\substack{i \in [t] \\ \{x,y\} \in E(G_i)}} B_i^{\{x,y\}}(F)$ *contains exactly one copy of every vertex in* $B$.

Proof. Let $j \in [|B|]$. The vertex $s_{\bar{p},\bar{q}}^{j} \in S_{\bar{p},\bar{q}}$ is endpoint of an edge $f$ in $F_S$. Let $v$ be the other endpoint of this edge, i.e., $f = \{s_{\bar{p},\bar{q}}^{j}, v\}$. Since $F_S \subseteq E(S, \bigcup_{i=1}^{t} \bigcup_{e \in E(G_i)} B_i^e)$ and $s_{\bar{p},\bar{q}}^{j}$ is only adjacent to vertices in $\bigcup_{i=1}^{t} \bigcup_{e \in E(G_i)} B_i^e$ that correspond to vertex $b_j$ in $B$, it holds that $v = b_{i^*,j}^{\{x,y\}}$ for some $i^* \in [t]$, $x \in V_{\bar{p}}$, $y \in V_{\bar{q}}$ with $\{x,y\} \in E(G_{i^*})$. Thus, $B_C$ contains at least one copy of $b_j$ for all $j \in [|B|]$.

Assume, $B_C$ contains at least two copies of a vertex $b_j$ in $B$, with $j \in [|B|]$. Let $i_1, i_2 \in [t]$, let $e_1 \in E(V_{\bar{p}}, V_{\bar{q}}) \cap E(G_{i_1})$, and let $e_2 \in E(V_{\bar{p}}, V_{\bar{q}}) \cap E(G_{i_2})$ such that $b_{i_1,j}^{e_1}$, $b_{i_2,j}^{e_2}$ are contained in $B_C$ and either $i_1 \neq i_2$ or $e_1 \neq e_2$. Since both vertices $b_{i_1,j}^{e_1}$ and $b_{i_2,j}^{e_2}$ are only adjacent to vertex $s_{\bar{p},\bar{q}}^{j} \in S_{\bar{p},\bar{q}}$, the set $F_S$ is not a matching, which is a contradiction and proves the claim. $\square$

Let $B_1, B_2, \ldots, B_h \subseteq B$ be the sets in $B$ that correspond to the non-empty sets in $\{B_i^{\{x,y\}}(F) \mid x \in V_{\bar{p}}, y \in V_{\bar{q}}, i \in [t] \colon \{x,y\} \in E(G_i)\}$. That is, for each non-empty set $B_i^{\{x,y\}}(F)$, with $i \in [t]$, there is a set $B_r = \{b_j \mid b_{i,j}^{\{x,y\}} \in B_i^{\{x,y\}}(F)\}$. It holds that the sets $B_1, B_2, \ldots, B_h$ are a partition of $B$ (Claim 10.9) and that $\sum_{r=1}^{h} \operatorname{cost}(B_r) \leq \operatorname{cost}(B)$ (inequality (10.1)). This implies that $h = 1$: Otherwise $B_1, B_2, \ldots, B_h$ would be a non-trivial partition of $B$ with $\operatorname{cost}(B) \geq \sum_{r=1}^{h} \operatorname{cost}(B_i)$ which implies that $B$ is not strongly beneficial (Proposition 10.6 item (13)).

Thus, there exists exactly one vertex $x \in V_{\bar{p}}$, exactly one vertex $y \in V_{\bar{q}}$, and exactly one $i^* \in [t]$ with the property that $\{x,y\} \in E(G_{i^*})$ and that $B_{i^*}^{\{x,y\}}(F)$ is not the empty set. Furthermore, it holds that $B_{i^*}^{\{x,y\}}(F) = B_{i^*}^{\{x,y\}}$ (Claim 10.9). It follows that for each graph $H_i^e$ with $i \in [t]$ and $e \in E(G_i)$ either $B_i^e(F) = B_i^e$ or $B_i^e(F) = \emptyset$. Therefore, for all $1 \leq p < q \leq k$ there exists exactly one edge $e \in E(V_p, V_q)$ and exactly one $i \in [t]$ such that $e \in E(G_i)$ and $B_i^e(F) = B_i^e$: Either all $1 \leq p < q \leq k$ fulfill inequation 10.1 with equality or there exist $1 \leq p' < q' \leq k$ such that inequation 10.1 holds with "$<$". This would imply that $B$ is not beneficial (see proof of Claim 10.9 and definition of strongly beneficial). Consequently, the edges in $F_S$ are incident with $\binom{k}{2}$ different copies of $H$ and cover the copy of $B$ in these $\binom{k}{2}$ copies.

Now, we consider which vertices are contained or not contained in $V(F_R)$. It holds that $|F_R| \leq \sum_{i=1}^{t} |E(G_i)| \cdot \text{EDS}(H) - \binom{k}{2}(\text{cost}(B) - |B|)$; to see this consider $k' \geq |F| = |F_S|+|F_T|+|F_Z|+|F_R|$. For each graph $H_i^e$, with $i \in [t]$ and $e \in E(G_i)$, with $B_i^e(F) = \emptyset$ we need at least $\text{EDS}(H)$ edges to dominate all edges in $E(H_i^e)$ and for each graph $H_i^e$, here $i \in [t]$ and $e \in E(G_i)$, with $B_i^e(F) = B_i^e$ we need at least $\text{EDS}(H - B)$ edges to dominate all edges in $E(H_i^e - B_i^e)$. Since no two different copies of $H$ are adjacent, since $\text{EDS}(H - B) = \text{EDS}(H) - |B| + \text{cost}(B)$, and since there are exactly $\binom{k}{2}$ copies of $H$ where the vertices that correspond to vertices in $B$ are covered by edges in $F_S$, it holds that we have exactly $\text{EDS}(H)$ or $\text{EDS}(H - B)$ edges of $F_R$ to dominate all edges in $H_i^e$ or $H_i^e - B_i^e$, respectively.

It follows, that $F_R$ contains no edge that has one endpoint in $W$ or $V$, because the vertex sets $V$ and $W$ are only adjacent to copies of vertices in $C$ in $G - X$ and a vertex $c \in C$ is neither extendable in $H$ nor in $H - B$. Hence, without using more than $\text{EDS}(H)$ or $\text{EDS}(H - B)$ edges to dominate all edges in $H$ or $H - B$, respectively, we cannot have an edge in $F_R$ that has one endpoint in $V \cup W$.

Let $i \in [t]$ and $e = \{x, y\} \in E(G_i)$ such that $B_i^e(F) = B_i^e$. It holds that $|F_i^e| = \text{EDS}(H - B)$, and therefore $C_i^e \not\subseteq V(F_i^e)$ (by definition of a control pair). Furthermore, the set $C_i^e \cap B_i^e$ is empty by the choice of $C$ and $B$. This implies that $N(C_i^e) = \{x, y\} \cup W(i) \subseteq V(F)$, i.e., that all neighbors of $C_i^e$ in $V$ and $W$ must be endpoints of $F$. It holds that $\{x, y\} \subseteq V(F_T)$ and $W(i) \subseteq V(F_Z)$ because neither edges in $F_S$ nor in $F_R$ have endpoints in $V \cup W$. Since, $|F_Z| = |W(i)|$ and $F_Z \subseteq E(Z, W)$ it follows that $V(F_Z) \cap W = W(i)$. Thus, all graphs $H_i^e$, here $i \in [t]$ and $e \in E(G_i)$, with $B_i^e(F) = B_i^e$ must belong to the same instance (because only edges in $F_Z$ contain edges that have endpoints in $W$ and because $F_Z$ has size $s = |W(i)|$).

Let $i^* \in [t]$ be the number of this instance and let $e = \{x, y\} \in E(G_{i^*})$ such that $B_{i^*}^e(F) = B_{i^*}^e$. It must hold that $x, y \in V(F_T)$ (because no other edges in $F$ have an endpoint in $V$). Since $|F_T| = k$ and $F_T \subseteq E(T, V)$ it follows that the $\binom{k}{2}$ graphs $H_{i^*}^e$, for $e \in E(G_{i^*})$, with $B_{i^*}^e(F) = B_{i^*}^e$ must correspond to a set of edges in $E(G_{i^*})$ that have their endpoints in the set $X = V(F_T) \cap V$ of size $k$. Consequently, the set $X$ is a clique in $G_{i^*}$ and $(G_{i^*}, k)$ is a yes-instance of Multicolored-Clique.

($\Leftarrow$:) This direction of the correctness proof is similar to the corresponding one in the proof of Theorem 9.1 and follows easily from the construction; we sketch this only briefly. If $(G_{i^*}, k)$ is a yes-instance and $X = \{x_1, \ldots, x_k\} \subseteq V$ is a multicolored $k$-clique in $G_{i^*}$ with $x_j \in V_j$, for all $j \in [k]$, then select $k + s$ solution edges for $F$ in $E(T, V)$ and $E(Z, W)$ as for Theorem 9.1. In particular, this ensures that $W(i^*) \cup X \subseteq V(F)$. For each set $S_{p,q}$ with $1 \leq p < q \leq k$ add the edges between the $d = |B|$ vertices between $S_{p,q}$ and the copy $B_{i^*}^{\{x_p,x_q\}}$ of $B$ in $H_{i^*}^{\{x_p,x_q\}}$. At this point, we have used up the budget (intuitively) intended for edges incident with $S$, $T$, and $Z$. All edges in $E(G[X'])$ are already dominated by $F$ as well as all edges incident with $S$; all edges in graphs $H_i^e$ and some edges between those graphs and $V \cup W$ remain.

In each graph $H_{i^*}^{\{x_p,x_q\}}$, with $1 \leq p < q \leq k$, we select an edge dominating set for $H_{i^*}^{\{x_p,x_q\}} - B_{i^*}^{\{x_p,x_q\}}$ of cost $\textsc{eds}(H - B)$. Together with previously added edges incident with $B_{i^*}^{\{x_p,x_q\}}$, this dominates all edges in this copy of $H$. Furthermore, edges between $H_{i^*}^{\{x_p,x_q\}}$ and $V \cup W$ are already dominated because their other endpoints are in $X \cup W(i^*) \subseteq V(F)$. For all other graphs $H_i^e$, i.e., with $i \neq i^*$ or with $i = i^*$ but $e \notin E(G[X])$ we can select an edge dominating set of size $\textsc{eds}(H)$ that is incident with $C_i^e$. This dominates all edges in this $H$-graph and, crucially, dominates all edges between $H_i^e$ and $V \cup W$ because their endpoints in $H_i^e$ are all in $C_i^e$. (This is the only place where we need the third property of control pairs.) Thus we have selected an edge dominating set and it can be readily checked that we have picked exactly $k' = s + k + \sum_{i=1}^{t} |E(G_i)| \cdot \textsc{eds}(H) + \binom{k}{2} \cdot \text{cost}(B)$. Note that $\binom{k}{2} \cdot \text{cost}(B)$ is exactly the additional cost of selecting $\binom{k}{2}$ times an edge dominating set for $H_{i^*}^{\{x_p,x_q\}} - B_{i^*}^{\{x_p,x_q\}}$ and $|B|$ edges between $B_{i^*}^{\{x_p,x_q\}}$ and $S_{p,q}$ rather than the optimum solution for $H_{i^*}^e$. ■

Theorem 10.8 implies that whenever we have a family of connected graphs $\mathcal{H}$ which contains at least one graph $H \in \mathcal{H}$ that has a control pair, then Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph does not have a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Now, one can ask which connected graphs have a control pair and whether Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph has a polynomial kernel when no graph in $\mathcal{H}$ has a control pair. First, we show which connected graphs have a control-pair. In a second step, we give a polynomial kernel for all remaining connected graphs of constant size.

### 10.3.1. Graphs that Contain Control Pairs

In the following four lemmata we show which properties of a graph $H$ help us to construct a control pair $(C, B)$ for $H$. We will see later that graphs that have no control pair have a very specific structure which helps us to obtain polynomial kernels for the remaining graphs.

**Lemma 10.10.** *Every connected graph $H$ that has at least one extendable vertex that is not free contains a control pair.*

**Proof.** Since graph $H$ has an extendable vertex that is not free, there exists a vertex $v \in Q(H) \setminus W(H)$ and a minimum edge dominating set $F$ in $H$ such that for every minimum edge dominating set $F'$ in $H - v$, which has size $|F| - 1$, it holds that $V(F) \setminus Q(H) \nsubseteq V(F')$: If there would be no such vertex $v \in Q(H) \setminus W(H)$, then it would hold that for all $v \in Q(H) \setminus W(H)$ and for all minimum edge dominating sets $F$ in $H$ there exists a minimum edge dominating set $F'$ in $H-v$ of size $|F|-1$ and with $V(F) \setminus Q(H) \subseteq V(F')$. This holds also for all vertices in $W(H)$ and would imply that $Q(H)$ is free. Now, let $v \in Q(H) \setminus W(H)$ be a vertex that is extendable but not free, and let $F$ be a minimum edge dominating set in $H$ such that there exists no minimum edge dominating set $F'$ (of size $|F|-1$) in $H-v$ with $V(F) \setminus Q(H) \subseteq V(F')$.

Let $C = V(F) \setminus Q(H)$ and let $B = \{v\}$. Since $B = \{v\} \subseteq Q(H)$ it holds that $C \subseteq V \setminus (Q(H) \cup B)$. We will show that $(C, B)$ is a control pair. The set $B$ is strongly beneficial, because $v$ is extendable and not free (Proposition 10.6 item (7)). Furthermore, by construction, it holds that $F$ is a minimum edge dominating set with $C \subseteq V(F)$ and that for all minimum edge dominating sets $F'$ in $H-B = H-v$ it holds that $C \nsubseteq V(F')$ (choice of $v$ and $C$).

It remains to show that no vertex in $C$ is extendable in $H-v$. Assume for contradiction that there exists a vertex $c \in C \cap Q(H-v)$. We will show that this implies that $c$ is also extendable in $H$, i.e., $c \in Q(H)$, which is a contradiction to the choice of $C$. Let $F_c$ be a minimum edge dominating set in $H-v-c$. Since $c$ is extendable in $H-v$, it holds that $|F_c| = \textsc{eds}(H-v-c) = \textsc{eds}(H-v)-1$. If $N_H(v) \setminus \{c\} = \emptyset$, then $F_c$ is also a minimum edge dominating set in $H-c$ and it would follow that $\textsc{eds}(H-c) \leq |F_c| = \textsc{eds}(H-v)-1 = \textsc{eds}(H)-2$; but $\textsc{eds}(H-c) \geq \textsc{eds}(H)-1$ (Proposition 10.6 item (4)), a contradiction. Thus, $N_H(v) \setminus \{c\} \neq \emptyset$ and we pick an arbitrary vertex $w \in N_H(v) \setminus \{c\}$. Now, $F_c \cup \{\{v,w\}\}$ would be an edge dominating set in $H-c$ of size $\textsc{eds}(H)-1$, hence $c$ would be extendable in $H$, which is a contradiction to the choice of $C$. ■

**Lemma 10.11.** *Every connected graph $H$ that has a strongly beneficial set, that contains at least one uncovered vertex, contains a control pair.*

**Proof.** Let $B$ be a strongly beneficial set in $H$ that contains at least one uncovered vertex and let $C = N_H(B \cap U(H)) \setminus B$ be the neighborhood of all uncovered vertices in $B$ without the vertices in $B$. Since $H$ is connected and every vertex in $U(H)$ has no neighbor in $U(H)$ or $Q(H)$ (Proposition 10.6 item (2)), it holds that $C \subseteq V(H) \setminus Q(H)$. Furthermore, $C$ is not the empty set. Otherwise, if all neighbors of $B \cap U(H)$ are also contained in $B$ then $\textsc{eds}(H-B) = \textsc{eds}(H-(B \setminus U(H)))$. This implies that $B$ is not beneficial which is a contradiction.

We will show that $(C, B)$ is a control pair. The set $B$ is strongly beneficial and the set $C$ is a subset of $V(H) \setminus (Q(H) \cup B)$ (by choice). It holds that every minimum edge dominating set in $H$ contains $C$, because $C \subseteq N(U(H))$ (Proposition 10.6 item (3)). Moreover, there exists no minimum edge dominating set $F'$ in $H-B$ such that $C \subseteq V(F')$: If not then such a set $F'$ would also be a minimum edge dominating set in $H-(B \setminus U(H))$ because $U(H) \cap B \subseteq U(H)$ is an independent set (Proposition 10.6 item (2)) and all neighbors of $U(H) \cap B$ are contained in $C$. But, this implies that $B$ is not beneficial because $\textsc{eds}(H-\widetilde{B}) \leq \textsc{eds}(H-B)$ where $\widetilde{B} = B \setminus U(H) \subsetneq B$.

Next, we show that no vertex in $C$ is extendable in $H-B$. Assume for the sake of contraction that there exists a vertex $c \in Q(H-B) \cap C$ that is extendable in $H-B$. Let

$F'$ be a minimum edge dominating set in $H - B - c$; hence $|F'| = \text{EDS}(H - B) - 1$. Since $c \in C = N_H(B \cap U(H))$, there exists a vertex $b \in B \cap U(H)$ with $\{c, b\} \in E(H)$. Now, $F' \cup \{\{c, b\}\}$ is an edge dominating set in $H - (B \setminus \{b\})$ of size $|F'| + 1 = \text{EDS}(H - B)$. Thus, $\text{EDS}(H - (B \setminus \{b\})) \leq \text{EDS}(H - B)$, which implies that $B$ is not beneficial, which is a contradiction and completes the proof. ∎

**Lemma 10.12.** *Every connected graph $H$ that contains at least one vertex that is not in $N[W(H)] \cup U(H)$ contains a control pair.*

**Proof.** We can assume that the graph $H$ neither contains an extendable vertex that is not free nor a strongly beneficial set which contains at least one uncovered vertex; otherwise we can apply Lemma 10.10 or Lemma 10.11, respectively, to find a control pair.

First, we prove that there exists a vertex $v \notin N[W(H)] \cup U(H)$ that is not contained in every minimum edge dominating set of $H$. Assume for contradiction that every vertex in $R := V(H) \setminus (N[W(H)] \cup U(H))$ is contained in every minimum edge dominating set of $H$. Let $v \in R$ and let $F$ be a minimum edge dominating set in $H$ with $|N_H(v) \cap V(F)|$ maximal. Let $x \in V(H)$ such that $\{v, x\} \in F$. Observe that $x \notin W(H)$, because $v \notin N[W(H)]$.

Consider the vertex set $X := N_H(v) \setminus V(F)$. It holds that $X$ neither contains a vertex of $R$ (because every vertex in $R$ is contained in every minimum edge dominating set) nor a vertex of $W(H)$ (because $v \notin N[W(H)]$). In addition, $X$ contains no vertex of $N(W(H))$: If $X$ would contain a vertex in $N(W(H))$ then we know that there exists an edge dominating set $F'$ in $H$ such that $(V(F) \cup N(W(H))) \setminus W(H) \subseteq V(F')$ (Proposition 10.6 item (6)). This implies that $|N_H(v) \cap V(F)| < |N_H(v) \cap V(F')|$ because $N_H(v) \cap V(F')$ contains all vertices that are contained in $N_H(v) \cap V(F)$ and the vertex in $N(W(H))$ that is contained in $X$. Recall that $W(H) \cap N_H(v) = \emptyset$. Furthermore, $X$ is not the empty set because this would imply that $x$ is extendable but not free: The set $F - \{\{x, v\}\}$ would be a minimum edge dominating set of $H - x$, because $N_H(v) \subseteq V(F)$. But, we assumed that $H$ contains no extendable vertex. Thus, the set $X$ is contained in $U(H)$ and not empty.

Let $Y = X \cup \{x\}$. The set $F - \{\{v, x\}\}$ is an edge dominating set in $H - Y$ because $F - \{v, x\}$ dominates all edges in $H - Y$ that are not adjacent to $v$ (the vertex $x$ is contained in $Y$), and vertex $v$ is only adjacent to the vertices in $Y$ and vertices that are contained in $V(F)$. Thus, it holds that $\text{EDS}(H - Y) < \text{EDS}(H)$. This implies that there exists a strongly beneficial set $B \subseteq Y$ (Proposition 10.6 item (11)). Since $x \notin W(H)$ and every extendable vertex is also free, no vertex in $Y$ is extendable and, therefore, $|B| \geq 2$ (Proposition 10.6 item (7) and (8)). This directly implies that $B \cap U(H) \neq \emptyset$ because $Y$ contains only one element, namely $x$, that is not in $U(H)$. This, however, is a contradiction since we assumed that $H$ contains no strongly beneficial set that contains at least one uncovered vertex. Thus, there exists a vertex in the set $R$ that is not contained in every minimum edge dominating set.

Let $v \in R$ be such a vertex that is not contained in every minimum edge dominating set and let $F$ be a minimum edge dominating set in $H$ that contains $v$ (such a minimum edge dominating set exists; otherwise $v$ would be uncovered). Let $x \in V(H)$ such that $\{v, x\} \in F$. The vertex $x$ is not contained in $W(H)$ because vertex $x$ is adjacent to vertex $v$, and because vertex $v$ is in $R$. Hence, neither vertex $v$ nor vertex $x$ are extendable because every extendable vertex is also free. The fact that $v, x \notin Q(H)$ together with Proposition 10.6 item (12) implies that the set $B = \{v, x\}$ is strongly beneficial. Let $C := N_H(v) \setminus \{x\}$. It holds that $C \cap Q(H)$ is empty because $W(H) = Q(H)$ and $v \notin N[W(H)]$. Furthermore, $C \cap B$ is empty (by choice of $C$).

Next, we will show that $(C, B)$ is a control pair. We already showed that $B$ is strongly beneficial. Recall that we chose a vertex $v \in R$ that is not contained in every minimum edge dominating set of $H$. Thus, there exists a minimum edge dominating set $F_v$ in $H$ that does not contain $v$. This edge dominating must contain all vertices in $N_H(v)$. Since $C$ is a subset of $N_H(v)$, there exists a minimum edge dominating set $\hat{F}$ in $H$ with $C \subseteq V(\hat{F})$. Furthermore, there exists no minimum edge dominating set $F_B$ in $H - B$ with $C \subseteq V(F_B)$: Otherwise, $x$ would be extendable (and not free) because $F_B$ would also be an edge dominating set in $H - x$. To prove that no vertex in $C$ is extendable in $H - B$ assume for the sake of contradiction that there exists a vertex $c \in C$ that is extendable in $H - B$. Thus, there exists a minimum edge dominating set $F_c$ in $H - B - c$ of size $\textsc{eds}(H - B) - 1$. We can extend $F_c$ to an edge dominating set in $H - c$ of size $\textsc{eds}(H - B) = \textsc{eds}(H) - 1$ by adding the edge $\{v, x\}$ to $F_c$. But now, $c$ is also extendable in $H$, which contradicts the assumption and shows that $(C, B)$ is a control pair. ■

So far, we showed that every connected graph $H$ that contains an extendable vertex that is not free, a vertex in $V(H) \setminus (N[W(H)] \cup U(H))$, or a strongly beneficial set that contains at least one vertex of the set $U(H)$ has a control pair. Thus, for all remaining connected graphs $H$ it holds that every extendable vertex is also free, i.e., $Q(H) = W(H)$, and that $V(H) \setminus (N[W(H)] \cup U(H))$ is empty. This implies that $V(H) = N[W(H)] \cup U(H)$. Moreover, no strongly beneficial set contains any vertex of $U(H)$ and, hence, strongly beneficial sets must be subsets of $N(W(H))$. Recall that (strongly) beneficial sets are subsets of $V(H) \setminus W(H)$.

**Lemma 10.13.** *Every connected graph $H$ that has a strongly beneficial set $B \subseteq N(W(H))$ such that no minimum edge dominating set $F_B$ of $H - B$ covers all vertices of $N(W(H)) \setminus B$ contains a control pair.*

**Proof.** We can assume that graph $H$ neither contains an extendable vertex that is not free, a strongly beneficial set which contains at least one uncovered vertex, nor a vertex that is not in $N[W(H)] \cup U(H)$; otherwise we can apply Lemma 10.10, Lemma 10.11 or Lemma 10.12, respectively. Thus $V(H) = N[W(H)] \cup U(H)$ and every strongly beneficial set is contained in $N(W(H))$.

Note that every strongly beneficial set $B \subseteq N(W(H))$ with the property that there exists no minimum edge dominating set in $H - B$ that contains all vertices of the set $N(W(H)) \setminus B$ together with the set $C = N(W(H)) \setminus B$ fulfills all except one property of a control pair: The set $B$ is strongly beneficial, $C \subseteq V(H) \setminus (Q(H) \cup B)$, and for every minimum edge dominating set $F_B$ in $H - B$ it holds that $C \nsubseteq V(F_B)$ (choice of $B$). Furthermore, there exists a minimum edge dominating set $F$ in $H$ such that $C \subseteq V(F)$, because $C \subseteq N(W(H))$ (Proposition 10.6 item (6)). But, we do not know whether $C \cap Q(H-B) = \emptyset$. We will show that if there exists a strongly beneficial set $B$ in $H$ such that no minimum edge dominating set in $H-B$ contains $N(W(H)) \setminus B$, then there exists also a strongly beneficial set $B'$ in $H$ such that no minimum edge dominating set in $H - B'$ contains $C' = N(W(H)) \setminus B'$, and such that $Q(H - B') \cap C' = \emptyset$.

Let $B \subseteq N(W(H))$ be a strongly beneficial set in $H$ such that no minimum edge dominating set of $H - B$ contains all vertices of the set $C := N(W(H)) \setminus B$, and with $|Q(H - B) \cap C|$ minimal under these strongly beneficial sets. (Such a set $B$ exists by assumption.) If $Q(H - B) \cap C = \emptyset$, then $B$ fulfills the desired property and $(C, B)$ is a control pair. Thus, assume that $|Q(H - B) \cap C| > 0$ and let $c \in Q(H - B) \cap C$. We show that $B' = B \cup \{c\}$ is a strongly beneficial set in $H$ that fulfills the same properties as $B$, and with $|Q(H - B) \cap C| > |Q(H - B') \cap C'|$, where $C' = N(W(H)) \setminus B'$, which is a contradiction to the choice of $B$.

First, we will show that $B' = B \cup \{c\}$ is a strongly beneficial set. It follows from Proposition 10.6 item (14) that there exists a partition $B_1, B_2, \dots, B_h$ of $B'$ where $B_i$ is either strongly beneficial or where $\text{cost}(B_i) = |B_i|$, for all $i \in [h]$, auch that $\text{cost}(B) \geq \sum_{i=1}^{h} \text{cost}(B_i)$ because $B \subseteq\subseteq N(W(H)) \subseteq V(H) \setminus W(H)$. We can assume, without loss of generality, that $c \in B_1$. Observe that $\text{cost}(B') = \text{cost}(B)$ because $c$ is extendable in $H - B$ which implies that $\text{EDS}(H - B') + 1 = \text{EDS}(H)$ (definition of cost). Now, if $h = 1$ and $B_1$ is strongly beneficial then $B' = B_1$ is strongly beneficial. Thus, assume for contraction that $h = 1$ and $|B_1| = \text{cost}(B_1)$ or that $h \geq 2$. If $h = 1$ and $\text{cost}(B_1) = |B_1|$ then it follows that $\text{cost}(B) = \text{cost}(B') = |B'| > |B|$, but it always holds that $\text{cost}(B) \leq |B|$ which is a contradiction. Thus, it holds that $h \geq 2$. Now, the sets $B_1 \setminus \{c\}, B_2, \dots, B_h$ are a partition of $B$ and it holds that

$$\text{cost}(B) = \text{cost}(B') \geq \sum_{i=1}^{h} \text{cost}(B_i) \geq \text{cost}(B_i \setminus \{c\} + \sum_{i=2}^{h} \text{cost}(B_i)$$

where the last inequality follows from Proposition 10.6 item (5). Since $B$ is strongly beneficial it follows that $h = 2$, $B_1 \setminus \{c\} = \emptyset$ and $B_2 = B$ is strongly beneficial. This implies that $\text{cost}(B_1) + \text{cost}(B_2) = 1 + \text{cost}(B) > \text{cost}(B)$ because $c$ is not extendable in $H$. But, this contradicts the assumption that $\text{cost}(B') \geq \text{cost}(B_1) + \text{cost}(B_2)$. This shows that $B'$ is strongly beneficial.

Now, we will show (by contradiction) that there exists no minimum edge dominating set $F_{B'}$ in $H - B'$ such that $C' := N(W(H)) \setminus B' \subseteq V(F_{B'})$. Assume that there exists a minimum edge dominating set $F_{B'}$ in $H - B'$ such that $C' \subseteq V(F_{B'})$. The set $F_B = F_{B'} \cup \{\{c, v\}\}$ with $v \in N_{H-B}(c)$ is a minimum edge dominating set of size $|F_{B'}| + 1 = \text{EDS}(H - B') + 1 = \text{EDS}(H - B)$ in $H - B$ because $F_B$ dominates all edges in $H - B'$ and all edges that are incident with $c$ (since $c \in V(F_B)$). Recall that $N_{H-B}(c)$ is not empty, because $c \in N(W(H))$ and $W(H) \cap B = \emptyset$ (definition of beneficial sets). But, the minimum edge dominating set $F_B$ in $H - B$ contains all vertices in

$C := N(W(H)) \setminus B$ as an endpoint: It holds $C = C' \cup \{c\}$ and $C' \subseteq V(F_{B'}) \subseteq V(F_B)$ as well as $c \in V(F_B)$. This contradicts the choice of $B$, hence there exists no minimum edge dominating set $F_{B'}$ in $H - B'$ such that $C' \subseteq V(F_{B'})$.

Finally, we show that $|Q(H-B) \cap C| > |Q(H-B') \cap C'|$ which contradicts the choice of the set $B$.

Claim 10.14. *It holds that $Q(H - B') \cap C' \subsetneq Q(H - B) \cap C$.*

Proof. ($\subseteq$:) If $Q(H - B') \cap C'$ is empty then $Q(H - B') \cap C' \subseteq Q(H - B) \cap C$ and we are done. Thus, assume that $Q(H - B') \cap C' \neq \emptyset$. Let $v \in Q(H - B') \cap C'$ be a vertex that is extendable in $H - B'$. This implies that there exists a minimum edge dominating set $F'_v$ in $H - B' - v$ of size $\textsc{eds}(H - B') - 1$. Consider $F_v := F'_v \cup \{\{c, u\}\}$ with $u \in N_{H-B-v}(c)$. Again, $N_{H-B-v}(c) \neq \emptyset$ because $c$ is adjacent to a vertex in $W(H)$ and $W(H) \cap (B' \cup \{v\}) = \emptyset$. The set $F_v$ is an edge dominating set (of size $\textsc{eds}(H-B') = \textsc{eds}(H-B)-1$) in $H-B-v$ because $F_v$ dominates all edges in $H-B'-v$ and all edges that are incident with $c$. Furthermore, $F_v$ is a minimum edge dominating set in $H-B-v$ because $|F_v| \geq \textsc{eds}(H-B-v) \geq \textsc{eds}(H-B)-1 = |F_v|$ (Proposition 10.6 item (4)). It follows that $\textsc{eds}(H - B - v) = \textsc{eds}(H - B) - 1$. Thus, $v \in Q(H - B)$ (definition of extendable vertices) which implies that $Q(H - B') \cap C' \subseteq Q(H - B) \cap C$.

($\neq$:) The vertex $c$ is extendable in $H - B$ (choice of $c$), i.e., $c \in Q(H - B) \cap C$. But, vertex $c$ is not extendable in $H - B'$, because $c \in B'$ and thus not contained in $H - B'$. This implies $Q(H - B) \cap C \neq Q(H - B') \cap C'$ and concludes the proof. $\square$

Now, $B'$ is a strongly beneficial set in $H$ such that no minimum edge dominating set $F_{B'}$ in $H - B'$ contains all vertices in $C'$. Furthermore, $Q(H-B') \cap C' \subsetneq Q(H-B) \cap C$ which contradicts the choice of $B$ because $|Q(H - B') \cap C'| < |Q(H - B) \cap C|$.

Overall, we showed that there exists a strongly beneficial set $B \subseteq N(W(H))$ such that there exists no minimum edge dominating set in $H - B$ that covers all vertices of $C := N(W(H)) \setminus B \subseteq V \setminus (Q(H) \cup B)$ and such that $Q(H - B) \cap C = \emptyset$. It holds that $(C, B)$ is a control pair (see argumentation above). $\blacksquare$

## 10.4. Generalizing the Kernelization

We showed for many finite sets $\mathcal{H}$ that Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph has no polynomial kernel, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Now, we will show that Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph admits a polynomial kernel for all remaining choices of finite sets $\mathcal{H}$. Recall that all remaining sets $\mathcal{H}$ only contain graphs $H$ where each vertex is either free, neighbor of a free vertex, or uncovered, i.e., $V(H) = N[W(H)] \cup U(H)$. Furthermore, it holds for every strongly beneficial set $B$ in $H$ that there exists a minimum edge dominating set $F_B$ in $H - B$ with $N(W(H)) \setminus B \subseteq V(F_B)$. Our goal is to generalize the kernels for Edge Dominating Set parameterized by the size of a modulator to a $P_5$-component graph (Section 9.4) and parameterized by the size of a degree-1-modulator (Section 9.3). The general kernel is more complicated because neither the $P_5$, the $P_1$, nor the $P_2$ have any strongly beneficial sets which are the main source of complication.

We first show that if no graph in $\mathcal{H}$ has a beneficial set then there exists a kernel with $\mathcal{O}(|X|^2)$ vertices and $\mathcal{O}(|X|^3)$ edges. (Recall, for every kernel lower bound we showed that there exists a strongly beneficial set, thus all graphs that have no beneficial set can only have vertices that are uncovered, free, or neighbors of free vertices.) This will later be extended to a more involved kernelization that also handles $H$-components where $H$ does have beneficial sets but they always have a minimum edge dominating set $F_B$ in $H - B$ as above. Afterwards, we will show that the kernel size is almost optimal in both cases.

### 10.4.1. Kernelization

To prove that Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph has a polynomial kernel, where $\mathcal{H}$ is a finite set of graph such that no graph in $\mathcal{H}$ contains a control pair, we construct a bipartite graph to mark certain connected components, as in the kernelization for Edge Dominating Set parameterized by the size of a modulator to a $P_5$-component graph. This marking becomes more complicated when a graph $H$ has one, or more, (strongly) beneficial sets. Besides marking connected components that allow us to cover vertices in $X$ without using $|X|$ extra edges we have to mark connected components that contain uncovered vertices similar to the kernelization for Edge Dominating Set parameterized by the size of a degree-1-modulator.

**Lemma 10.15.** *Let $\mathcal{H}$ be a finite set of connected graphs that contain no beneficial sets and such that each graph $H \in \mathcal{H}$ has $V(H) = N[W(H)] \cup U(H)$, i.e., each graph $H$ has only vertices that are uncovered, free, or neighbors of a free vertex. Then* Edge Dominating Set *parameterized by the size of a modulator $X$ to an $\mathcal{H}$-component graph admits a kernel with $\mathcal{O}(|X|^2)$ vertices, $\mathcal{O}(|X|^3)$ edges, and size $\mathcal{O}(|X|^3 \log |X|)$.*

**Proof.** Let $(G, k, X)$ be an instance of Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph. We can assume, without loss of generality, that $k - \textsc{eds}(G - X) < |X|$. Otherwise, we can return a trivial solution consisting of a minimum edge dominating set in $G - X$ and one edge in $\delta_G(x)$ for each vertex $x \in X$. Let $\mathcal{C}$ be the set of connected components in $G - X$, let $W$ be the set of all free vertices in $G - X$, and let $U$ be the set of all uncovered vertices in $G - X$, hence $W = \bigcup_{C \in \mathcal{C}} W(C)$ and $U = \bigcup_{C \in \mathcal{C}} U(C)$. Let $X_W \subseteq X$ be the set of vertices in $X$ that are adjacent to a vertex in $W$, hence $X_W = \{x \in X \mid \exists w \in W \colon \{x, w\} \in E(G)\}$, and let $\mathcal{C}_W$ be the set of connected components $C$ in $\mathcal{C}$ where $W(C)$ is adjacent to a vertex in $X_W$, hence with $E(W(C), X_W) \neq \emptyset$.

To find vertices in $X$ that can be covered by every edge dominating set of size at most $k$ without spending extra budged we construct a bipartite graph $G_W$. One part of $G_W$ is the set $X_W$ and the other part consists of one vertex $s_C$ for each connected component $C$ in $\mathcal{C}_W$. We add an edge between a vertex $x \in X_W$ and a vertex $s_C$ with $C \in \mathcal{C}_W$ if and only if vertex $x$ is adjacent to a vertex in $W(C)$ in $G$. Now, we apply Theorem 2.2 to obtain either a maximum matching in $G_W$ that saturates $X_W$, or to find a set $Y \subseteq X_W$ such that $|N_{G_W}(Y)| < |Y|$ and such that there exists a maximum matching in $G_W - N_{G_W}[Y]$ that saturates $X_W \setminus Y$. If there exists a maximum matching in $G_W$ that saturates $X_W$ then let $X_W^h = X_W$, let $X_W^l = \emptyset$, and let $\mathcal{C}_W^l = \emptyset$. Otherwise, if there exists a set $Y$ with the above properties then let $X_W^l = Y$, let $X_W^h = X_W \setminus Y$, and let $\mathcal{C}_W^l$ be the set of connected components $C$ in $\mathcal{C}_W$ where $W(C)$ is adjacent to a vertex in $X_W^l$, i.e., $\mathcal{C}_W^l = \{C \in \mathcal{C}_W \mid s_C \in N_{G_W}(X_W^l)\}$. It holds that $|\mathcal{C}_W^l| = |N_{G_W}(Y)| < |Y| = |X_W^l|$ because every connected component in $\mathcal{C}_W^l$ corresponds to a vertex in $N_{G_W}(Y)$. In both cases it holds that $|\mathcal{C}_W^l| \leq |X_W^l|$.

**Reduction Rule 10.1.** Delete the set $X_W^h$ from $G$, i.e., let $G' = G - X_W^h$, $X' = X \setminus X_W^h$ and $k' = k$.

Claim 10.16. *Reduction Rule 10.1 is safe.*

Proof. Let $F$ be an edge dominating set of size at most $k$ in $G$. We construct an edge dominating set $F'$ of size at most $k' = k$ in $G'$ as follows. First, we delete every edge $e \in F$, if both endpoints of $e$ are contained in $X_W^h$. Next, for every edge $e = \{x, y\} \in F$ that has exactly one endpoint in $X_W^h$ (without loss of generality $x \in X_W^h$) we either replace $e$ by one edge in $\delta_{G'}(y)$, if $\delta_{G'}(y) \neq \emptyset$ or delete $e$, if $\delta_{G'}(y) = \emptyset$. It holds that $F'$ has size at most $k = k'$ because we only delete edges or replace edges. Furthermore, $V(G') \setminus V(F')$ is an independent set because $V(F')$ contains every vertex in $V(F) \setminus X_W^h$ that is not isolated in $G'$, because $V(G) \setminus V(F)$ is an independent set in $G$, and because $V(G') = V(G) \setminus X_W^h$. Thus, $F'$ is an edge dominating set of size at most $k'$ in $G'$.

For the other direction, let $F'$ be an edge dominating set of size at most $k'$ in $G'$ that is also a matching. Let $M$ be a maximum matching in $G_W - N_{G_W}[X_W^l]$ that saturates $X_W^h$, and for each vertex $x \in X_W^h$ let $C_x$ be the connected component in $\mathcal{C}_W$ with $\{s_C, x\} \in M$. Consider the connected component $C_x$ for a vertex $x \in X_W^h$. Since $M$ is a maximum matching in $G_W - N_{G_W}[X_W^l]$ it holds that the connected component $C_x$ does not correspond to a vertex in $N_{G_W}(X_W^l)$. This implies that no free vertex in $C_x$ is adjacent to a vertex in $X_W^l$ (construction of $G_W$). Let $F'_x$ be the set of edges in $F'$ that have at least one endpoint in $C_x$, i.e., $F'_x = \{f \in F' \mid f \cap C_x \neq \emptyset\}$. We partition $F'_x$ in three sets. Let $F'_{x,1}$ be the set of edges in $F'_x$ that have one endpoint in $X$ and the other endpoint in $C_x$, let $F'_{x,2}$ be the set of edges in $F'_x$ that have one endpoint in $U(C_x)$ and the other endpoint in $N(W(C_x))$, and let $F'_{x,3}$ be the set of remaining edges in $F'_x$. Recall that uncovered vertices are only adjacent to vertices in $N(W(C_x)) \cup X$ (Proposition 10.6 item (2)). Thus, all edges in $F'_x$ that have one endpoint in $U(C_x)$ (the set of uncovered vertices in $C_x$) are contained in $F'_{x,1} \cup F'_{x,2}$. Let $B_x$ be the set of vertices in $C_x$ that are incident with an edge in $F'_{x,1} \cup F'_{x,2}$, hence $B_x = V(C_x) \cap V(F'_{x,1} \cup F'_{x,2})$. Recall that every vertex in $W(C_x)$ is only adjacent to vertices in $N[W(C_x)]$ in $G'$ because no free vertex of $C_x$ is adjacent to a vertex in $X_W^l$ (and we delete set $X_W^h$ to obtain $G'$). Thus, $V(F'_{x,1} \cup F'_{x,2})$ contains no free vertex of

$C_x$ because no free vertex in $C_x$ is adjacent to a vertex in $X$ or an uncovered vertex of $C_x$; thus $B_x \subseteq N(W(C_x)) \cup U(C_x)$. Since $C_x$ has no beneficial set, it holds that $\textsc{eds}(C_x - B_x) = \textsc{eds}(C_x)$ (Proposition 10.6 item (9)). Thus, the set $F'_x$ has size at least $\textsc{eds}(C_x)$ plus the size of $|F'_{x,1} \cup F'_{x,2}|$. Furthermore, $F'_{x,3}$ is an edge dominating set in $C_x - B_x$ (by choice of $B_x$) which implies that $|F'_{x,3}| \geq \textsc{eds}(C_x - B_x) = \textsc{eds}(C_x)$. It follows that $|F'_x| = |F'_{x,1} \cup F'_{x,2}| + |F'_{x,3}| \geq |F'_{x,1} \cup F'_{x,2}| + \textsc{eds}(C_x)$.

Let $w_x \in N_G(x) \cap W(C_x)$ be a free vertex in $C_x$ that is adjacent to $x \in X^h_W$. We replace $F'_{x,3}$ by the set $F_{x,3}$ that consists of a minimum edge dominating set in $C_x - w_x$ that covers all vertices in $N(W(C_x))$ (which exists by Proposition 10.6 item (6)) and the edge $\{w_x, x\}$. The edge set $F_{x,3}$ has size $\textsc{eds}(C_x)$ because $w_x \in W(C_x)$ (definition of free); hence $|F_{x,3}| \leq |F'_{x,3}|$. We do this for all vertices $x \in X^h_W$ to obtain $F$. It holds that $F$ has size at most $|F'|$ because we only replace $F'_{x,3}$ by $F_{x,3}$ for each $x \in X^h_W$, and because $|F_{x,3}| \leq |F'_{x,3}|$.

It remains to prove that $F$ is indeed an edge dominating set in $G$. The set $V(F)$ contains all vertices in $V(F')$, except some free vertices in the connected components where we change the edge dominating set. But, all neighbors of these free vertices are contained in $V(F)$, because these free vertices are only adjacent to vertices in $X^h_W$, which are contained in $V(F)$, and to vertices in the connected component of $G - X$ they belong to. Thus, $F$ is an edge dominating set in $G$. $\square$

Now, let $X_U \subseteq X$ be the set of vertices in $X$ that are adjacent to a vertex in $U$, i.e., $X_U = \{x \in X \mid \exists u \in U \colon \{x, u\} \in E(G)\}$. We partition $X_U$ into two sets as follows. Let $X^h_U \subseteq X_U$ be the set of vertices in $X_U$ that are adjacent to the set of uncovered vertices in at least $|X| + 1$ connected components, i.e.,

$$X^h_U = \{x \in X_U \mid \exists C_1, C_2, \ldots, C_{|X|+1} \in \mathcal{C} \text{ and } \forall j \in [|X| + 1] \colon x \in N_G(U(C_j))\},$$

and let $X^l_U = X_U \setminus X^h_U$ be the set of vertices in $X_U$ that are adjacent to the set of uncovered vertices in less than $|X|$ connected components in $\mathcal{C}$. By $\mathcal{C}^l_U$ we denote the connected components $C$ in $\mathcal{C}$ with $E(U(C), X^l_U) \neq \emptyset$. It holds that $|\mathcal{C}^l_U| \leq |X^l_U| \cdot |X|$.

**Reduction Rule 10.2.** For all $x \in X^h_U$ add a vertex $x'$ and the edge $\{x, x'\}$ to $G$.

Let $\mathcal{C}_S$ be the set of new connected components in $G'$. These are the connected components consisting of a single vertex $x'$ that we add during Reduction Rule 10.2. It is easy to verify that Reduction Rule 10.2 is safe, i.e. that there exists a solution for $(G, k, X)$ if and only if there exists a solution for $(G', k', X')$. This follows from the fact that every edge dominating set $F$ in $G$ of size at most $k$ must contain the vertices in $X^h_U$ as endpoints: If there would be a vertex $x \in X^h_U$ that is not contained in $V(F)$, then there exist at least $|X| + 1$ connected components that contain a vertex in $U$ that is adjacent to $x$. Now, these at least $|X| + 1$ vertices must be contained in $V(F)$. But, every connected component $C$ with this property is adjacent to $\textsc{eds}(C) + 1$ edges in $F$ because no minimum edge dominating set in $C$ covers an uncovered vertex. Thus, $|F| \geq \textsc{eds}(G - X) + |X| + 1 > k$ because two connected components in $G - X$ are not adjacent. This is a contradiction and shows that every vertex in $X_U^h$ must be contained in an edge dominating set of size at most $k$ in $G$. By adding the vertex $x'$ and the edge $\{x, x'\}$, with $x \in X_U^h$, to $G$, we encode that vertex $x$ must be an endpoint of an edge in every solution of size at most $k$.

Let $\mathcal{C}_D \subseteq \mathcal{C}$ be the set of connected components in $G - X$ that are not contained in $\mathcal{C}_U^l \cup \mathcal{C}_W^l \cup \mathcal{C}_S$. Note, $\mathcal{C}_D$ contains all connected components where neither a free nor an uncovered vertex is adjacent to a vertex in $X$.

**Reduction Rule 10.3.** Delete all connected components in $\mathcal{C}_D$ and decrease $k$ by the size of a minimum edge dominating set in $\mathcal{C}_D$, i.e., let $G' = G - \mathcal{C}_D$, $X' = X$, and $k' = k - \textsc{eds}(\mathcal{C}_D)$.

Claim 10.17. *Reduction Rule 10.3 is safe.*

Proof. Let $F$ be an edge dominating set of size at most $k$ in $G$. If no edge in $F$ has one endpoint in a connected component of $\mathcal{C}_D$ and the other endpoint in $X$, then $F' = F \setminus E(\mathcal{C}_D)$ is an edge dominating set of size at most $|F| - \textsc{eds}(\mathcal{C}_D) \leq k'$ in $G'$ and we are done. Thus, assume that there exists at least one edge that has one endpoint in $X$ and one endpoint in a connected component of $\mathcal{C}_D$; hence $F \cap E(V(\mathcal{C}_D), X) \neq \emptyset$. We denote the set of edges in $F$ that are incident with a connected component in $\mathcal{C}_D$ by $F_D$, and let $F_{D,X} = F \cap E(V(\mathcal{C}_D), X)$.

It holds that every vertex $v$ in $V(F_{D,X})$ is not a free vertex: Otherwise, there exists a vertex $x \in X_W$ such that $\{x, v\} \in F_{D,X}$. But, we delete every vertex in $X_W^h$ during Reduction Rule 10.1 and every connected component that contains a free vertex that is adjacent to a vertex in $X_W^l$ is a connected component in $\mathcal{C}_W^l$ and therefore not in $\mathcal{C}_D$.

Let $\widetilde{X} \subseteq X$ be the set of vertices in $X$ that are contained in $V(F_D)$, i.e., $\widetilde{X} = X \cap V(F_D)$. Since no vertex in $\widetilde{X}$ is adjacent to a free vertex in $\mathcal{C}_D$, and since the connected components do not have beneficial sets it holds that the size of a minimum edge dominating set in $\mathcal{C}_D$ plus the size of the set $\widetilde{X}$ is smaller or equal to the number of edges in $F_D$. Now, to obtain an edge dominating set $F'$ of size at most $k'$ in $G'$, we delete the edge set $F_D$ from $F$ and we add for all vertices $x \in \widetilde{X}$ exactly one edge of the set $\delta_{G'}(x)$ to the edge dominating set (or none if the edge set $\delta_{G'}(x)$ is empty). By construction it follows that $F'$ has size at most $k'$ because we delete $|F_D| \geq \textsc{eds}(\mathcal{C}_D) + |\widetilde{X}|$ edges from $F$ and add at most $|\widetilde{X}|$ edges to $F$. Furthermore, $F'$ is an edge dominating set in $G'$, because $V(F')$ contains all vertices of $V(F)$ that are contained in $V(G')$ and not isolated in $G'$.

For the other direction, let $F'$ be an edge dominating set of size at most $k'$ in $G'$. To obtain an edge dominating set $F$ of size at most $k$ in $G$ we add for every connected component $C \in \mathcal{C}_D$ a minimum edge dominating set $F_C$ in $C$ with $N(W(C)) \subseteq V(F_C)$ to $F'$ (the existence of such a minimum edge dominating set follows from Proposition 10.6 item (6)). To show that $F$ is indeed an edge dominating set in $G$ we only have to show that every edge $e \in E(\mathcal{C}_D, X)$ is dominated by $F$. All other edges are dominated by $F$, because they are already dominated by $F' \subseteq F$ or by $F_C$ with $C \in \mathcal{C}_D$. Assume for contradiction that there exists a connected component $C \in \mathcal{C}_D$ and an edge $e = \{v, x\}$

in $G$ with $v \in V(C)$ and $x \in X$ such that neither vertex $v$ nor vertex $x$ is an endpoint of an edge in $F$. Since all vertices in $N(W(C))$ are endpoints of an edge in $F$ (choice of $F_C$), and since $C$ contains only free vertices, neighbors of free vertices and uncovered vertices, it holds that $v$ is either a free or uncovered vertex in $C$; hence $x$ is contained in $X_W \cup X_U$. It holds that all vertices in $X_U^h$ are contained in $V(F') \subseteq V(F)$ because every vertex $x$ in $X_U^h$ is adjacent to a connected component in $\mathcal{C}_S$ that consists of a single vertex $x'$ which is only adjacent to vertex $x$ in $X_U^h$. Furthermore, during Reduction Rule 10.1 we delete the vertex set $X_W^h$. Thus, $x$ is contained in $X_U^l \cup X_W^l$. Now, if $v$ is a free vertex in $C$, then $x$ must be a vertex in $X_W^l$ which implies that $C$ is a connected component in $\mathcal{C}_W^l$ and not in $\mathcal{C}_D$, which is a contradiction. Similar, if $v$ is an uncovered vertex in $C$, then $x$ must be a vertex in $X_U^l$ which implies that $C$ is a connected component in $\mathcal{C}_U^l$ and not in $\mathcal{C}_D$, which is a contradiction. This shows that $F$ is an edge dominating set of size at most $k$ in $G$. □

We already showed that each reduction rule is safe. Next, we show that the reduced instance $(G', k', X')$ has at most $\mathcal{O}(|X|^2)$ vertices. The set of connected components in $G' - X'$ is $\mathcal{C}' := \mathcal{C}_U^l \cup \mathcal{C}_W^l \cup \mathcal{C}_S$, because we only add connected components to $G$ during Reduction Rule 10.2 (namely the components in $\mathcal{C}_S$) and we only delete connected components during Reduction Rule 10.3. We delete all connected components that are not contained in $\mathcal{C}_U^l \cup \mathcal{C}_W^l \cup \mathcal{C}_S$. It follows that $G' - X'$ has at most $2|X|^2$ connected components, because $|\mathcal{C}'| \leq |\mathcal{C}_U^l| + |\mathcal{C}_W^l| + |\mathcal{C}_S| \leq |X_U^l| \cdot |X| + |X_W^l| + |X_U^h| \leq 2|X|^2$. Since every connected component has constant size, and since $V(G') = V(\mathcal{C}') \cup X'$ it holds that $G'$ has at most $\mathcal{O}(|X|^2)$ vertices. Next, we have to bound the number of edges. Every connected component has only constant size, thus it has only a constant number of edges (because our graph is simple); hence $|E[\mathcal{C}']| \in \mathcal{O}(|X|^2)$. The number of edges between vertices in $X$ is at most $|X|^2$. All remaining edges are between $X$ and $\mathcal{C}'$ and there are at most $|X| \cdot |V(\mathcal{C}')| \in \mathcal{O}(|X|^3)$ edges between $X$ and $\mathcal{C}'$. This sums up to at most $\mathcal{O}(|X|^3)$ edges.

It is easy to see that we can perform the reduction in polynomial time: We apply every Reduction Rule exactly once and we also compute every set exactly once. Furthermore, we can compute the sets $W$ and $U$ in polynomial time because we can compute a minimum edge dominating set in a connected component of constant size in constant time. Moreover, we can compute all remaining sets in polynomial time (by applying Theorem 2.2 or by simple counting). We can also apply each Reduction Rule in polynomial time because we only delete respectively add vertex or edge sets of size polynomial in $|G|$ which we can compute in polynomial time. ■

Remark 10.18. If $\mathcal{H}$ is a finite set of connected graphs that contain neither beneficial sets nor uncovered vertices (and such that for each graph $H \in \mathcal{H}$ it holds $V(H) = N[W(H)]$) then Edge Dominating Set parameterized by the size of a modulator $X$ to an $\mathcal{H}$-component graph admits a kernel with $\mathcal{O}(|X|)$ vertices, $\mathcal{O}(|X|^2)$ edges, and size $\mathcal{O}(|X|^2 \log |X|)$. This holds because the only set that has size $\mathcal{O}(|X|^2)$ is the set of connected components in $\mathcal{C}_U^l$. But, if we have no uncovered vertex then $\mathcal{C}_U^l = \emptyset$. Hence, the reduced instance has only $\mathcal{O}(|X|)$ many connected components, and therefore, only $\mathcal{O}(|X|)$ many vertices.

**Lemma 10.19.** *Let $d \in \mathbb{N}$ and let $\mathcal{H}$ be a finite set of connected graphs such that no graph $H \in \mathcal{H}$ has a strongly beneficial set of size exceeding $d$, such that $V(H) = N[W(H)] \cup U(H)$ for all $H \in \mathcal{H}$, and such that each strongly beneficial set $B$ of any graph $H \in \mathcal{H}$ is contained in $N(W(H))$. Moreover, assume that for each strongly beneficial set $B$ of a graph $H \in \mathcal{H}$ there exists a minimum edge dominating set in $H - B$ that covers all vertices in $N(W(H)) \setminus B$. Then* Edge Dominating Set *parameterized by the size of a modulator $X$ to an $\mathcal{H}$-component graphs admits a kernel with $\mathcal{O}(|X|^d)$ vertices, $\mathcal{O}(|X|^{d+1})$ edges, and size $\mathcal{O}(|X|^{d+1} \log |X|)$.*

**Proof.** Let $(G, k, X)$ be an instance of Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph. Again, we can assume, without loss of generality, that $k - \text{EDS}(G - X) < |X|$ (see proof of Lemma 10.15). The kernelization is similar to the previous kernelization. We construct a different graph $G_W$ to compute $X^h_W$ because we have to be a little bit more careful with connected components in $\mathcal{C}^l_U$ and because we have beneficial sets. Furthermore, we define the set $\mathcal{C}_D$ differently (for this purpose we compute another auxiliary graph).

In the previous kernelization (Lemma 10.15) the connected components in $G - X$ have no beneficial set. Thus, for every connected component $C$ that has at least one free vertex and whose set of free vertices is adjacent to at least one vertex in $X$, we could assume that an edge dominating set contains at least one of these edges between the free vertices and $X$. We cannot assume this anymore because a connected component can also contain a beneficial set $B$. Now, it could be necessary that an edge dominating set of size at most $k$ contains a matching between $B$ and $X$. For example, assume that there exists a connected component $C$ that contains a free vertex that is adjacent to vertex $x_1$ in $X$, and that $C$ has also a beneficial set $B = \{b_1, b_2, b_3\}$ of size three with $\text{cost}(B) = 1$ such that $b_i$ is adjacent to a vertex $x_i$ in $X$ (where all $x_i$ are pairwise different). Now, we can either increase the cost locally by one to cover three vertices in $X$ or use the same local cost to cover one vertex, namely $x_1$, in $X$. Further, assume that $x_1, x_2, x_3$ are endpoints of edges in every edge dominating set of size at most $k$ and that there exist no way to cover $x_2$ and $x_3$ with an extra budget of one. Hence, the edge dominating set will probably contain the edges between $B$ and $\{x_1, x_2, x_3\}$ as well as a minimum edge dominating set in $C - B$, and no edge between a free vertex of $C$ and $X$.

Besides this, in the previous kernelization we could assume that every edge that is incident with an uncovered vertex increases the cost locally by one. We cannot assume this anymore because a connected component $C$ with a beneficial set $B$ can cover $|B|$ vertices in $N(B) \cap U(C)$ while increasing the cost locally by $\text{cost}(B) < |B|$. Thus, a beneficial set $B$ could be useful to cover some vertices in $U(C)$ and to cover vertices in $X$. But, if all vertices in $U(C)$ are only adjacent to vertices in $X$ that must be in every edge dominating set of size at most $k$ then we will never cover the vertices in $U(C)$ by edges inside a connected component.

For these reasons, we compute the set $X_W^h$ using a different auxiliary graph which leads to $\mathcal{O}(|X|^2)$ connected components in $\mathcal{C}_W^l$ (in the worst case). Note that it would be possible to bound the number of connected components in $\mathcal{C}_W^l$ by $\mathcal{O}(|X|)$ by defining an auxiliary graph that handles the connected components that have free vertices or beneficial sets at the same time. This would make the analysis more complicated. But, we are only able to bound the number of connected components that contain a beneficial set by $\mathcal{O}(|X|^d)$, with $d \geq 2$. Thus, even if no graph has uncovered vertices we will not be able to reduce to less than $\mathcal{O}(|X|^2)$ vertices because we have connected components that contain beneficial sets (of size $d \geq 2$). Recall, in the previous kernelization we can reduce to $\mathcal{O}(|X|)$ vertices if we have no uncovered vertices.

As before, let $\mathcal{C}$ be the set of connected components in $G - X$, let $W$ be the set of all free vertices in $G - X$, and let $U$ be the set of all uncovered vertices in $G - X$. Let $X_W \subseteq X$ be the set of vertices in $X$ that are adjacent to a vertex in $W$, hence $X_W = \{x \in X \mid \exists w \in W \colon \{x, w\} \in E(G)\}$, and let $\mathcal{C}_W$ be the set of connected components $C$ in $\mathcal{C}$ where $W(C)$ is adjacent to a vertex in $X_W$, hence with $E(W(C), X_W) \neq \emptyset$. Again, we compute a bipartite graph $G_W$: One part consists of $|X|$ vertices $x^1, x^2, \ldots, x^{|X|}$ for every vertex $x$ in $X_W$. We denote this set by $R$. The other part consists of one vertex $s_C$ for every connected component $C$ in $\mathcal{C}_W$. We add an edge between a copy $x^i$ of vertex $x \in X_W$, with $i \in [|X|]$, and a vertex $s_C$ with $C \in \mathcal{C}_W$ if and only if $x$ is adjacent to a vertex in $W(C)$. Now, we apply Theorem 2.2 to obtain either a maximum matching in $G_W$ that saturates $R$, or to find a set $Y \subseteq R$ such that $|N_{G_W}(Y)| < |Y|$ and such that there exists a maximum matching in $G_W - N_{G_W}[Y]$ that saturates $R \setminus Y$. Observe, since every copy of a vertex $x \in X_W$ has the same neighborhood it holds that either all copies of $x$ are contained in $Y$ or none. If there exists a maximum matching in $G_W$ that saturates $R$ then let $X_W^h = X_W$, let $X_W^l = \emptyset$, and let $\mathcal{C}_W^l = \emptyset$. Otherwise, if there exists a set $Y$ with the above properties then let $X_W^l = \{x \in X_W \mid x^1 \in Y\}$ be the vertices in $X$ whose copies are contained in $Y$, let $X_W^h = X_W \setminus X_W^l$, and let $\mathcal{C}_W^l$ be the set of connected components $C$ in $\mathcal{C}_W$ where $W(C)$ is adjacent to a vertex in $X_W^l$. Note that every connected component $C$ in $\mathcal{C}_W^l$ corresponds to a vertex in $N_{G_W}(Y)$. Thus, the set $\mathcal{C}_W^l$ contains at most $|N_{G_W}(Y)| < |Y| = |X| \cdot |X_W^l|$ connected components. Now, we apply Reduction Rule 10.1.

Claim 10.20. *Reduction Rule 10.1 is safe.*

Proof. Let $F$ be an edge dominating set of size at most $k$ in $G$. We can construct an edge dominating set of size at most $k' = k$ in $G'$ as in the proof of Claim 10.16. We delete every edge $e = \{x, y\} \in F$ if $x, y \in X_W^h$ or if $x \in X_W^h$ and $y$ is isolated in $G'$. Furthermore, we replace each edge $e = \{x, v\} \in F$ with $x \in X_W^h$ and $v \notin X_W^h$ (not isolated in $G'$) by an edge in $\delta_{G'}(v)$. By construction, the resulting set $F'$ is an edge dominating set of size at most $k = k'$ in $G'$.

For the other direction, let $F'$ be an edge dominating set of size at most $k'$ in $G'$. Recall that $k - \textsc{eds}(G - X) < |X|$, which implies that $k' - \textsc{eds}(G' - X') < |X|$ because $\textsc{eds}(G' - X') = \textsc{eds}(G - X)$ and $k = k'$. Thus, there are at most $|X| - 1$ connected components $C$ of $G' - X'$ that are incident with more than $\textsc{eds}(C)$ edges of $F'$. Recall, the graph $G_W - N_{G_W}[Y]$ (respectively the graph $G_W$ if $X_W^l = \emptyset$) contains a matching $M$ that saturates $R \setminus Y = \{x^i \mid x \in X_W^h \wedge i \in [|X|]\}$. For every vertex $x \in X_W^h$ let $C_x^1, C_x^2, \ldots, C_x^{|X|}$ be the connected components in $\mathcal{C}_W \setminus \mathcal{C}_W^l$ with $\{x^i, s_{C_x^i}\} \in M$. Note that the set of free vertices in these connected components is not adjacent to a vertex in $X_W^l$ because all connected components whose set of free vertices is adjacent to a vertex in $X_W^l$ correspond to a vertex in $N_{G_W}(Y)$.

Since at most $|X|-1$ connected components $C$ of $G'-X'$ are incident with more than $\textsc{eds}(C)$ edges of $F'$ at least one of the connected components $C_x^1, C_x^2, \ldots, C_x^{|X|}$ is only incident with $\textsc{eds}(C_x^i)$ edges of $F'$. Say, without loss of generality, that for all $x \in X_W^h$ the connected component $C_x^1$ is only incident with $\textsc{eds}(C_x^1)$ edges of $F'$. (Note that for two different vertices $x, y \in X_W^h$ the connected components $C_x^1$ and $C_y^1$ are different.) Furthermore, the set of free vertices in $C_x^1$ is only adjacent to vertices in $C_x^1$ because we delete $X_W^h$ to obtain $G'$ and every connected component whose set of free vertices is adjacent to a vertex in $X_W^l$ is contained in $\mathcal{C}_W^l$. Since we have only $\textsc{eds}(C_x^1)$ edges to dominate all edges in $C_x^1$ it holds that no edge has an endpoint in $U(C_x^1)$ or one endpoint in $C_x^1$ and the other endpoint in $X$. Let $w_x \in W(C_x^1)$ be a free vertex in $C_x^1$ that is adjacent to vertex $x$ in $G$; hence $\{w_x, x\} \in E(G)$. Now, for every $x \in X_W^h$ we delete all edges that are incident with $C_x^1$ from $F'$ and add a minimum edge dominating set in $C_x^1 - w_x$ that covers $N(W(C_x^1))$ (Proposition 10.6 (6)) as well as the edge $\{w_x, x\}$ to obtain $F$. It holds that $F$ has size $k$ because $\textsc{eds}(C_x^1) = \textsc{eds}(C_x^1 - w_x) + 1$. It remains to prove that $F$ is an edge dominating set in $G$. The set $V(F)$ contains all vertices in $V(F')$ except some free vertices in the connected components where we change the edge dominating set. But, all neighbors of these vertices are contained in $V(F)$ because these free vertices are only adjacent to vertices in $X_W^h$ (which are contained in $V(F)$) and to vertices in the connected component of $G - X$ they belong to. Thus, $F$ is an edge dominating set of size at most $k$ in $G$. □

Let $X_U \subseteq U$ be the set of vertices in $X$ that are adjacent to a vertex in $U$, let $X_U^h \subseteq X_U$ be the set of vertices in $X_U$ that are adjacent to the set of uncovered vertices in at least $|X|+1$ connected components, and let $X_U^l = X_U \setminus X_U^h$. Again, by $\mathcal{C}_U^l$ we denote the connected components $C$ in $\mathcal{C}$ with $E(U(C), X_U^l) \neq \emptyset$, and it holds that $|\mathcal{C}_U^l| \leq |X| \cdot |X_U^l|$.

Next, we apply Reduction Rule 10.2. Let $\mathcal{C}_S$ be the set of new connected components in $G - X$ that we add during Reduction Rule 10.2. To prove that Reduction Rule 10.2 is safe in the previous kernelization (Lemma 10.15), we only showed that every vertex in $X_U^h$ must be covered by every solution of size at most $k$. The same argumentation holds here. Thus, Reduction Rule 10.2 is safe.

Let $\mathcal{C}_B$ be the set of connected components $C$ in $\mathcal{C} \setminus (\mathcal{C}_W^l \cup \mathcal{C}_U^l \cup \mathcal{C}_S)$ that contain a strongly beneficial set. To find connected components in $\mathcal{C}_B$ that can be safely removed from $G$ we construct an auxiliary graph $G_A$ as follows:

- Add for each set $Y \subseteq X$ with $2 \le |Y| \le d$ and for each $\beta \in [|Y| - 1]$ a vertex $r_{Y,\beta}$ to $G_A$; denote the union of these vertices by $R$.
- Add for each connected component $C$ in $\mathcal{C}_B$ a vertex $s_C$ to $G_A$; denote the union of these vertices by $S$.
- For each connected component $C \in \mathcal{C}_B$ we add the edge $\{r_{Y,\beta}, s_C\}$ to $G_A$ if and only if there exists a strongly beneficial set $B$ of size $|Y|$ in $C$ with $\text{cost}(B) = \beta$ such that there exists a perfect matching $M$ in $(Y \cup B, E(G) \cap E(Y, B))$. Hence, we add an edge between a vertex $s_C$ that represents connected component $C$ and a vertex $r_{Y,\beta}$ if and only if a local solution for $C$ that contains a maximum matching between $Y$ and $B$ (that covers $Y$) increases the cost of a local solution for $C$ only by $\beta$.

Thus, the auxiliary graph tells us which connected components $C \in \mathcal{C}_B$ can help us to cover a set $Y \subseteq X$ (of size at most $d$) without using $|Y|$ "additional" edges (or more precisely by using $\beta$ "additional" edges). Observe that $G_A$ is bipartite with bipartition $R$ and $S$.

Claim 10.21. *We can construct graph $G_A$ in polynomial time.*

Proof. The set $R$ has $\sum_{i=2}^{d} \binom{|X|}{i} \cdot i \in \mathcal{O}(|X|^d)$ many vertices, one for each pair $(Y, \beta)$ where $Y \subseteq X$ with $2 \le |Y| \le d$ and $\beta \in [|Y|]$. Hence we can construct $R$ in polynomial time. Since the set $S$ contains one vertex for each connected component in $\mathcal{C}_B$ we can construct $S$ in polynomial time.

For every connected component $C \in \mathcal{C}_B$ we can compute a minimum edge dominating set in $C$ as well as in $C - B$, for every $B \subseteq V(C)$, in polynomial time because $C$ is of constant size. Thus, we can compute all strongly beneficial sets $B$ in $C$ as well as the value $\text{cost}(B)$ in polynomial time: There are only a constant number of possible sets and we can compute $\textsc{eds}(C)$ and $\textsc{eds}(H - B)$ in polynomial time. This is only possible, because $C$ is of constant size. Let $C$ be a connected component in $\mathcal{C}_B$, let $B$ be a strongly beneficial set in $C$, and let $Z = N(B) \cap X$ be the set of vertices in $X$ that are adjacent to a vertex in $B$. For each set $Y \subseteq Z$ of size $|B|$ we add the edge $\{r_{Y,\text{cost}(B)}, s_C\}$ to $G_A$, if there exists a perfect matching in $(B \cup Y, E(B, Y))$. We can do this in polynomial time, because we only have to compute $\binom{|Z|}{|B|} \le |X|^d$ many maximum matchings. We can do this for every connected component and every beneficial set in a connected component because there are only a polynomial number of connected components and because every connected component has only a constant number of strongly beneficial sets. $\square$

Now, we apply Theorem 2.2 to obtain either a maximum matching in $G_A$ that saturates $R$, or to find a set $Z \subseteq R$ such that $|N_{G_A}(Z)| < |Z|$ and such that there exists a maximum matching in $G_A - N_{G_A}[Z]$ that saturates $R \setminus Z$. Note that if there exists a matching in $G_A$ that saturates $R$ we can set $Z = \emptyset$. Thus, we can assume that we always find a set $Z$ that fulfills the above properties. Let $M$ be a maximum matching in $G_A - N_{G_A}[Z]$. By choice of $Z$, it holds that the matching $M$ saturates $R \setminus Z$. Now, let $\mathcal{C}_B^l$ be the set of connected components $C$ in $\mathcal{C}_B$ that correspond to a vertex $s_C$ in $N_{G_A}(Z)$, and let $\mathcal{C}_B^h = \{C \in \mathcal{C}_B \mid \exists Y \subseteq X, \beta \in [|Y|-1] \colon \{r_{Y,\beta}, s_C\} \in M\}$ be the set of connected components $C$ in $\mathcal{C}_B$ that correspond to a vertex $s_C$ in $S$ with the property that $\{r_{Y,\beta}, s_C\}$ is an edge in $M$ for a set $Y \subseteq X$ and an integer $\beta \in [|Y|-1]$. Note that $r_{Y,\beta} \in R \setminus Z$. Now, we can bound the number of connected components in $\mathcal{C}_B^l$ and $\mathcal{C}_B^h$: It holds that $|\mathcal{C}_B^l| = |N_{G_A}(Z)| < |Z|$ (property of $Z$) and that $|\mathcal{C}_B^h| = |R \setminus Z|$ (because we add one vertex for every vertex in $R \setminus Z$ to $\mathcal{C}_B^h$); thus $|\mathcal{C}_B^l \cup \mathcal{C}_B^h| \leq |R| \in \mathcal{O}(|X|^d)$.

So far, we know that $|\mathcal{C}_W^l| \leq |X| \cdot |X_W^l|$, that $|\mathcal{C}_U^l| \leq |X| \cdot |X_U^l|$, that $|\mathcal{C}_S| \leq |X_U^h|$, and that $|\mathcal{C}_B^l \cup \mathcal{C}_B^h| \in \mathcal{O}(|X|^d)$. We will show that the remaining connected components of $G - X$ can be safely removed by reducing the value $k$ accordingly. Let $\mathcal{C}_D$ be the set of connected components in $G - X$ that are not contained in $\mathcal{C}_U^l \cup \mathcal{C}_S \cup \mathcal{C}_W^l \cup \mathcal{C}_B^l \cup \mathcal{C}_B^h$. We apply Reduction Rule 10.3 to delete all connected components in $\mathcal{C}_D$ and to obtain our reduced instance.

Claim 10.22. *Reduction Rule 10.3 is safe.*

Proof. Let $(G, k, X)$ be the instance before applying Reduction Rule 10.3, and let $(G', k', X')$ be the instance after applying Reduction Rule 10.3. Note that $X = X'$, because we only delete connected components of $G - X$ and decrease $k$.

Let $F'$ be an edge dominating set of size at most $k'$ in $G'$. To obtain an edge dominating set of size at most $k$ in $G$, we add for each connected component $C$ in $\mathcal{C}_D$ a minimum edge dominating set $F_C$ with $N(W(C)) \subseteq V(F_C)$ to $F'$. The existence of such a minimum edge dominating set follows from Proposition 10.6 item (6). We denote the resulting set by $F$. By construction, the set $F$ has size at most $k$ because $k' = k - \textsc{eds}(\mathcal{C}_D)$ and we add exactly a minimum edge dominating set of $\mathcal{C}_D$ to $F'$ to obtain $F$. To show that $F$ is an edge dominating set of $G$, we have to show that $F$ dominates every edge between $X$ and a connected component $C$ in $\mathcal{C}_D$: All other edges are either dominated by $F'$ or by the added minimum edge dominating sets $F_C$ with $C \in \mathcal{C}_D$. Assume for contradiction that there exists a connected component $C \in \mathcal{C}_D$, a vertex $v \in V(C)$, and a vertex $x \in X$ such that $\{v, x\} \in E(G)$ is not dominated by $F$. All vertices in $N(W(C))$ are incident with an edge in $F_C$ (choice of $F_C$) and therefore incident with an edge in $F$. Thus, $v$ must be a free or uncovered vertex because $V(C) = N[W(C)] \cup U(C)$. If $v$ is a free vertex, then $x$ must be contained in $X_W$. Since we delete $X_W^h$ during Reduction Rule 10.1, it holds that $x \in X_W^l$. But this implies that $C$ is a connected component in $\mathcal{C}_W^l$ because $\mathcal{C}_W^l$ contains all connected components whose set of free vertices is adjacent to a vertex in $X_W^l$. Thus, $C$ is contained in $\mathcal{C}_W^l$ and not in $\mathcal{C}_D$ which is a contradiction. If $v$ is an uncovered vertex then $x$ must be a vertex in $X_U$. All vertices in $X_U^h$ must be incident with an edge in $F'$ because every vertex $x$ in $X_U^h$ is adjacent to a connected component in $\mathcal{C}_S$ that consists of a single vertex $x'$ which is only adjacent to this vertex $x$ in $X$; hence $x \in X_U^l$. But, this implies that $C$ is contained in $\mathcal{C}_U^l$, and not in $\mathcal{C}_D$, because every connected component whose set of uncovered vertices is adjacent to a vertex in $X_U^l$ is contained in $\mathcal{C}_W^l$. This is a contradiction. Thus, $F$ is an edge dominating set of size at most $k$ in $G$.

For the other direction, let $F$ be an edge dominating set of size at most $k$ in $G$ that is also a matching. Recall, every vertex $x \in X$ that is adjacent to a vertex $v \in W(C)$ is contained in $X_W^l$ because we delete $X_W^h$ during Reduction Rule 10.1. Hence $x \in X_W^l$ which implies that $C \in \mathcal{C}_W^l$. Thus, every connected component $C$ with $E(W(C), X) \neq \emptyset$ is contained in $\mathcal{C}_W^l$, and therefore, not contained in $\mathcal{C}_D \cup \mathcal{C}_B$.

Let $C_1, C_2, \dots, C_p$ be all connected components in $\mathcal{C}_D \cup \mathcal{C}_B^h$ that are incident with an edge in $F$ that has its other endpoint in the set $X$. Thus, for all $i \in [p]$ it holds that $E(C_i, X) \cap F \neq \emptyset$. For each connected component $C_i$, with $i \in [p]$, let $B_i = \{c \in V(C_i) \mid \exists x \in X \colon \{x, c\} \in F\}$ be the set of vertices in $V(C_i)$ that are incident with an edge in $F$ that has its other endpoint in $X$. Note that $B_i \subseteq V(C_i) \setminus W(C_i)$ because $E(W(C_i), X) = \emptyset$ for all connected components in $\mathcal{C}_D \cup \mathcal{C}_B$. Since $B_i \subseteq V(C) \setminus W(C)$ we can apply Proposition 10.6 (14): For all $i \in [p]$ let $B_i^1, B_i^2, \dots, B_i^{q_i} \subseteq B_i$ be a partition of $B_i$ where $B_i^j$ is either strongly beneficial or has $\mathrm{cost}(B_i^j) = |B_i^j|$, for all $j \in [q_i]$, such that $\mathrm{cost}(B_i) \geq \sum_{j=1}^{q_i} \mathrm{cost}(B_i^j)$. Let $Y_i^j = \{x \in X \mid \exists v \in V(B_i^j) \colon \{x, v\} \in F\}$, with $i \in [p]$ and $j \in [q_i]$, be the set of vertices in $X$ that are incident with an edge in $F$ whose other endpoint is contained in $B_i^j$, and let $\beta_i^j = \mathrm{cost}(B_i^j)$. Note that the sets $B_i^j$ and $Y_i^j$ have the same size because $F$ is a matching in $G$. Therefore, if $B_i^j$ is a strongly beneficial set then the size of $Y_i^j$ is at most $d$ because every strongly beneficial set has size at most $d$.

Now, if $B_i^j$, with $i \in [p]$ and $j \in [q_i]$, is a strongly beneficial set in $C_i$ then it holds that $2 \leq |B_i^j| = |Y_i^j| \leq d$ and that $\beta_i^j = \mathrm{cost}(B_i^j) < |B_i^j| = |Y_i^j|$. Thus, the vertex $r_{Y_i^j, \beta_i^j}$ is contained in $R$. Note that $C_i$ is a connected component in $\mathcal{C}_B \setminus \mathcal{C}_B^l$ if $B_i^j$ is strongly beneficial for an index $j \in [q_i]$ because $C_i \in \mathcal{C}_D \cup \mathcal{C}_B^h$, $\mathcal{C}_B \subseteq \mathcal{C} \setminus (\mathcal{C}_W^l \cup \mathcal{C}_U^l \cup \mathcal{C}_S)$ and $\mathcal{C}_D = \mathcal{C} \setminus (\mathcal{C}_W^l \cup \mathcal{C}_U^l \cup \mathcal{C}_S \cup \mathcal{C}_B^l \cup \mathcal{C}_B^h)$. Furthermore, the vertex $r_{Y_i^j, \beta_i^j}$ must be contained in $R \setminus Z$: If $r_{Y_i^j, \beta_i^j}$ is a vertex in $Z$ then $N_{G_A}(r_{Y_i^j, \beta_i^j}) \subseteq N_{G_A}(Z)$. But, every vertex in $N_{G_A}(Z)$ corresponds to a connected component in $\mathcal{C}_B^l$ which would imply that $C_i$ is contained in $\mathcal{C}_B^l$ because $r_{C_i} \in N_{G_A}(r_{Y_i^j, \beta_i^j})$ which is a contradiction. This implies that $r_{Y_i^j, \beta_i^j} \in R \setminus Z$.

Let $C_i^j$ be the connected component in $\mathcal{C}_B^h$ whose corresponding vertex in $S$ is matched to $r_{Y_i^j, \beta_i^j}$; hence with $\{r_{Y_i^j, \beta_i^j}, s_{C_i^j}\} \in M$. Since $\{r_{Y_i^j, \beta_i^j}, s_{C_i^j}\} \in E(G_A)$ there exists a strongly beneficial set $\bar{B}_i^j$ of size $|Y_i^j|$ in $C_i^j$ with $\mathrm{cost}(\bar{B}_i^j) = \beta_i^j (= \mathrm{cost}(B_i^j))$ such that there exists a perfect matching between $Y_i^j$ and $\bar{B}_i^j$ in $(Y_i^j \cup \bar{B}_i^j, E(G) \cap E(Y_i^j, \bar{B}_i^j))$. Hence, for every strongly beneficial set $B_i^j$, with $i \in [p]$ and $j \in [q_i]$, the set $Y_i^j$ is associated with a different connected component $C_i^j$ in $\mathcal{C}_B^h$ and a beneficial set $\bar{B}_i^j$ that has the same advantage as $B_i^j$.

In the case that the size of $B_i^j$, with $i \in [p]$ and $j \in [q_i]$, is equal to $\mathrm{cost}(B_i^j)$ it holds that $Y_i^j$ is equal to $\beta_i^j$. Thus, every edge that is incident with a vertex in $Y_i^j$ increases the cost locally by one.

To construct an edge dominating set of size at most $k'$ in $G'$ we delete all edges in $F$ that are incident with a vertex in a connected component of $\mathcal{C}_D \cup \mathcal{C}_B^h$; denote the resulting set by $\widetilde{F}$. Recall that $C_1, C_2, \ldots, C_p$ are the connected components in $\mathcal{C}_D \cup \mathcal{C}_W^h$ that are incident with an edge in $F$ whose other endpoint is contained in $X$. Next, for each $i \in [p]$ and $j \in [q_i]$ with $B_i^j \subseteq B_i$ strongly beneficial we add a minimum edge dominating set in $C_i^j - \bar{B}_i^j$ with $N(W(C_i^j)) \setminus \bar{B}_i^j \subseteq V(F_C)$ to $\widetilde{F}$ as well as a maximum matching between $\bar{B}_i^j$ and $Y_i^j$ that saturates both sets: Such a minimum edge dominating set exists by assumption and such a matching exists because $\{r_{Y_i^j, \beta_i^j}, s_{C_i^j}\}$ is an edge in $G_A$. Recall that every connected component $C_i^j$ is contained in $\mathcal{C}_B^h$. Thus, all added edges are contained in $G'$. For all remaining connected components $C$ in $\mathcal{C}_B^h$ we add a minimum edge dominating set in $C$ to $\widetilde{F}$ that contains $N(W(C))$. Finally, we add for each vertex $y$ that is contained in a set $Y_i^j$ with $|Y_i^j| = |B_i^j| = \mathrm{cost}(B_i^j) = \beta_i^j$, where $i \in [p]$ and $j \in [q_i]$, an arbitrary edge in $\delta_{G'}(y)$ to $\widetilde{F}$ if $\delta_{G'}(y)$ is not the empty set. Otherwise, if $v$ is isolated in $G'$ we add no edge to $\widetilde{F}$. We denote the resulting set by $F'$.

First, we show that $F'$ is indeed an edge dominating set of $G'$. Every vertex in $X$ that is incident with an edge in $F$ that has its other endpoint in a connected component of $\mathcal{C}_D \cup \mathcal{C}_B^h$ is contained in a set $Y_i^j$, with $i \in [p]$ and $j \in [q_i]$. During the reduction we delete only edges that are incident with a connected component in $\mathcal{C}_D \cup \mathcal{C}_B^h$, but we also add for each vertex $v$ in $Y_i^j$ that is not isolated in $G'$ an edge in $\delta_{G'}(y)$ to $F'$. Thus, every vertex in $X$ that is covered by $V(F)$ and not isolated in $G'$ is also covered by $V(F')$. Furthermore, every edge in a connected component is dominated: We delete only edges that are incident with connected components in $\mathcal{C}_D \cup \mathcal{C}_B^h$ and we add for all connected components in $\mathcal{C}_B^h$ an edge dominating set to $F'$. Since the connected components in $\mathcal{C}_D$ are not contained in $G'$ it holds that $F'$ dominates all edges in $G' - X'$. Thus, the only edges that are possibly not dominated by $F'$ have one endpoint in $X$ and the other endpoint in a connected component of $\mathcal{C}_B^h$ because these are the only connected components where we change the edge dominating set (and because $V(F')$ covers all vertices in $X \cap V(F)$ that are not isolated in $G'$). Assume that there exists a connected component $C \in \mathcal{C}_B^h$, a vertex $v \in V(C)$, and a vertex $x \in X$ such that edge $\{v, x\} \in E(G')$ is not dominated by $F'$; hence $v, x \notin V(F')$. No vertex in $W(C)$ is adjacent to a vertex in $X$ (otherwise $C \in \mathcal{C}_W^l$ and not in $\mathcal{C}_B^h$). Furthermore, every vertex in $N(W(C))$ is incident with an edge in $F'$ (by construction). Thus, $v$ must be a vertex in $U(C)$, and $x$ must be a vertex in $X_U$. Every vertex in $X_U^h$ must be incident with an edge in $F'$ because $X_U^h \subseteq V(F)$, and because every vertex in $X$ that is covered by the edge dominating set $F$ and is not isolated in $G'$ is contained in $V(F')$; hence $x \in X_U^l$. But, if $x \in X_U^l$ then it follows that $C$ is a connected component in $\mathcal{C}_U^l$ and not in $\mathcal{C}_B^h$, which is a contradiction; thus $F'$ is an edge dominating set in $G'$.

It remains to show that $F'$ contains at most $k'$ edges. The connected component $C_i$, with $i \in [p]$, is incident with at least $\textsc{eds}(C_i) + \mathrm{cost}(B_i)$ many edges (definition of cost). For all remaining connected components $C \in \mathcal{C}_D \cup \mathcal{C}_B^h$ we need at least $\textsc{eds}(C)$ many edges. Thus, it holds that $\widetilde{F} \leq |F| - \textsc{eds}(\mathcal{C}_D \cup \mathcal{C}_B^h) - \sum_{i=1}^{p} \sum_{j=1}^{q_i} \beta_i^j$ because

$$|\widetilde{F}| \leq |F| - \left(\textsc{eds}(\mathcal{C}_D \cup \mathcal{C}_B^h) + \sum_{i=1}^{p} \operatorname{cost}(B_i)\right) \leq |F| - \textsc{eds}(\mathcal{C}_D \cup \mathcal{C}_B^h) - \sum_{i=1}^{p} \sum_{j=1}^{q_i} \operatorname{cost}(B_i^j)$$
$$\leq |F| - \textsc{eds}(\mathcal{C}_D \cup \mathcal{C}_B^h) - \sum_{i=1}^{p} \sum_{j=1}^{q_i} \beta_i^j$$

To obtain $F'$ we added $\textsc{eds}(C_i^j) + \operatorname{cost}(\bar{B}_i^j) = \textsc{eds}(C_i^j) + \beta_i^j$ edges to $\widetilde{F}$, with $i \in [p]$, and $j \in [q_i]$, if $B_i^j \subseteq B_i$ is a strongly beneficial set in $C_i$: We add a minimum edge dominating set in $C_i^j - \bar{B}_i^j$ as well as a matching between $Y_i^j$ and $\bar{B}_i^j$ that saturates both sets to $\widetilde{F}$, and $\operatorname{cost}(\bar{B}_i^j) = \beta_i^j$. All these connected components $C_i^j$ are contained in $\mathcal{C}_B^h$. For all remaining connected components $C$ in $\mathcal{C}_B^h$ we add a minimum edge dominating set of $C$ to $\widetilde{F}$ that contains $N(W(C))$. Furthermore, for all vertices $y$ that are not isolated in $G'$ and that are contained in a set $Y_i^j$, with $i \in [p]$, and $j \in [q_i]$, where $|Y_i^j| = \beta_i^j = \operatorname{cost}(B_i^j)$, we add an arbitrary edge in $\delta_{G'}(y)$ to $\widetilde{F}$. Thus,

$$|F'| \leq |\widetilde{F}| + \textsc{eds}(\mathcal{C}_B^h) + \sum_{i=1}^{p} \sum_{j=1}^{q_i} \beta_i^j \leq |F| - \textsc{eds}(\mathcal{C}_D) = k'.$$

This completes the proof. □

We showed that all reduction rules are safe. To show that the reduced $(G', k', X')$ instance has only $\mathcal{O}(|X|^d)$ vertices, we only have to bound the number of connected components in $G' - X'$ because every connected component has constant size. During the reduction rules we delete all connected components that are not contained in $\mathcal{C}' := \mathcal{C}_W^l \cup \mathcal{C}_U^l \cup \mathcal{C}_S \cup \mathcal{C}_B^h \cup \mathcal{C}_B^l$. We already showed that $|\mathcal{C}_W^l \cup \mathcal{C}_U^l \cup \mathcal{C}_S| \leq 2 \cdot |X|^2$ (see above). Furthermore, we showed that $|\mathcal{C}_B^l \cup \mathcal{C}_B^h| \in \mathcal{O}(|X|^d)$. This implies that $G'$ has at most $\mathcal{O}(|X|^d)$ connected components, and thus, at most $\mathcal{O}(|X|^d)$ vertices. (We assumed that there exists at least one graph $H$ in $\mathcal{H}$ that has a beneficial set and this beneficial set has at least size two; thus $d \geq 2$.) Next, we have to bound the number of edges. Every connected component has only constant size, thus it has only a constant number of edges; hence $|E[\mathcal{C}']| \in \mathcal{O}(|X|^d)$. The number of edges between vertices in $X$ is at most $|X|^2$. All remaining edges are between $X$ and $\mathcal{C}'$ which are at most $|X| \cdot |V(\mathcal{C}')| \in \mathcal{O}(|X|^{d+1})$ many edges. In total, this sums up to at most $\mathcal{O}(|X|^{d+1})$ edges.

It remains to show that we can perform the reduction in polynomial time. We can compute the sets $W$ and $U$ in polynomial time because every connected component is of constant size, and therefore, we can compute minimum edge dominating sets in every connected component as well as in every subgraph of a connected component in polynomial time. Furthermore, we can construct the auxiliary graphs $G_W$ and $G_A$ in polynomial time. Hence, by applying Theorem 2.2 we can compute the set $X_W^h$, the set $X_W^l$, the set $\mathcal{C}_W^l$, the set $\mathcal{C}_B^l$, and the set $\mathcal{C}_B^h$ in polynomial time. The set $X_U^l$, the set $X_U^h$, as well as the set $\mathcal{C}_U^l$ can be computed in polynomial time by simple counting. Since we can compute all sets in polynomial time, we can apply the reduction rules in polynomial time because we only delete the set $X_W^h$ (Reduction Rule 10.1) as well as all connected components that are not contained in $\mathcal{C}_U^l \cup \mathcal{C}_S \cup \mathcal{C}_W^l \cup \mathcal{C}_B^l \cup \mathcal{C}_B^h$ (Reduction Rule 10.3), and we only add one vertex for every vertex in $X_U^h$ (Reduction Rule 10.2). ■

### 10.4.2. Lower Bound on the Kernel Size

We have seen in the previous section that Edge Dominating Set parameterized by the size of certain $\mathcal{H}$-component graphs admits a polynomial kernel where the exponent depends on the class $\mathcal{H}$. More precisely, the exponent is two, when no graph in $\mathcal{H}$ contains a beneficial set, and $d+1$, when the largest strongly beneficial set of a graph $H$ in $\mathcal{H}$ is $d$. We will show that it is necessary that the kernel size depends exponentially on $d$.

**Theorem 10.23.** *Let $d \in \mathbb{N}$ and let $\mathcal{H}$ be a finite set of connected graphs such that some $H \in \mathcal{H}$ has a strongly beneficial set of size $d$. Then* Edge Dominating Set *parameterized by the size of a modulator to $\mathcal{H}$-component graphs does not have a kernelization of size $\mathcal{O}(|X|^{d-\varepsilon})$, for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly.

At a first glance, Exact $d$-dimension $d$-Set Cover[3] seems to be a suitable problem to prove Theorem 10.23 by giving a polynomial parameter transformation from Exact $d$-dimension $d$-Set Cover parameterized by the size of the universe to Edge Dominating Set parameterized by the size of a modulator to $\mathcal{H}$-component graphs, where $\mathcal{H}$ contains a graph $H$ that has a strongly beneficial set $B$ of size $d$. Indeed, if the beneficial set $B = \{b_1, b_2, \dots, b_d\}$ has for example $\text{cost}(B) = 1$ then there exists an easy polynomial parameter transformation from Exact $d$-dimension $d$-Set Cover parameterized by the size of the universe to Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph, where $\mathcal{H}$ contains a graph $H$ that has a strongly beneficial set $B = \{b_1, b_2, \dots, b_d\}$ of size $d$: For an instance $(U := U_1 \dot\cup U_2 \dot\cup \dots \dot\cup U_d, \mathcal{F})$ of Exact $d$-dimension $d$-Set Cover we construct an instance $(G, k, X)$ of Edge Dominating Set by adding for each element $u \in U$ two vertices $u$ and $u'$ to $G$ as well as an edge between them. Additionally, for each set $F = \{u_1, u_2, \dots, u_d\} \in \mathcal{F}$, where $u_i \in U_i$ for all $i \in [d]$, we add a copy $H_F$ of $H$ to $G$ and we add an edge between $u \in U$ and the copy of $b_i$ in $H_F$ if $u \in U_i \cap F$. Let $k = \text{EDS}(G - X) + \frac{|U|}{d}$ and let $X$ be the set of all vertices that are not contained in a copy of $H$, i.e., $X = \{u, u' \mid u \in U\}$. In general, if $\text{cost}(B) > 1$ then one would set $k = \text{EDS}(G - X) + \frac{|U|}{d} \cdot \text{cost}(B)$.

However, there could be cases where it seems unlikely that such a polynomial parameter transformation exists, and where the above construction is not correct. For

[3]In the Exact $d$-dimension $d$-Set Cover problem, we are given a universe $U = U_1 \dot\cup U_2 \dot\cup \dots \dot\cup U_d$, a set $\mathcal{F} \subseteq \binom{U}{d}$ such that for all $F \in \mathcal{F}$ it holds that $|F \cap U_i| = 1$, and the objective is to find a set $\mathcal{F}' \subseteq \mathcal{F}$ such that each element $u \in U$ is contained in exaclty one set of $\mathcal{F}'$.

example, assume that there exists a connected graph $H$ that has a strongly beneficial set $B = \{b_1, b_2, \ldots, b_{15}\}$ of size 15 with $\mathrm{cost}(B) = 10$. It could be possible that also the sets $B_1 = \{b_1, b_2, \ldots, b_9\}$, $B_2 = \{b_3, b_4, \ldots, b_{12}\}$, $B_3 = \{b_7, b_8, \ldots, b_{15}\}$, $B_4 = \{b_1, b_2, b_3, b_{10}, b_{11}, \ldots, b_{15}\}$, and $B_5 = \{b_1, b_2, \ldots, b_6, b_{13}, b_{14}, b_{15}\}$ are strongly beneficial sets of size 9 and that $\mathrm{cost}(B_i) = 6$ for all $i \in [5]$. Note that this does not violate the definition of strongly beneficial sets.

Now, there are two possibilities to cover 45 vertices in the modulator by using only edges between the modulator $X$ and the connected components of $G - X$ and using 30 edges more than one need to cover these copies. The first possibility is to use edges between the modulator $X$ and three copies of $B$ in different copies of $H$. Since $\mathrm{cost}(B) = 10$ and $|B| = 15$ we cover 45 vertices in $X$ by using 30 edges more than we need to cover these three components. The second possibility it to use edges between the modulator $X$ and copies of $B_1$, $B_2$, $B_3$, $B_4$ and $B_5$ in different copies of $H$. Since for all $i \in [5]$ it holds that $\mathrm{cost}(B_i) = 6$ and $|B_i| = 9$ it follows that we cover 45 vertices in $X$ by using 30 edges more that we need to cover these five copies of $H$. The problem is that in the second case, we cannot transform an edge dominating set in $G$ of size at most $\textsc{eds}(G - X) + \frac{|U|}{d} \cdot \mathrm{cost}(B)$ into an exact set cover of $(U, \mathcal{F})$. We can handle this problem by giving a cross-composition of cost $t^{1/d}$ from the $\mathsf{NP}$-hard Multicolored Clique problem.

**Proof of Theorem 10.23.** Fix a graph $H \in \mathcal{H}$ that contains a strongly beneficial set of size $d$, and fix a strongly beneficial set $B = \{b_1, \ldots, b_d\}$ of size $d$ in $H$. If any of the items 1a through 1d of Theorem 10.4 applies to $H$ then we already ruled out *any* polynomial kernelization (unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$). Thus, it suffices to prove the theorem in the remaining case where we know that $V(H) = N[W(H)] \cup U(W)$, that $B \subseteq N(W(H))$, and that there is a minimum edge dominating set $F_B$ of $H - B$ that covers $N(W(H)) \setminus B$.

To prove the theorem, we give a cross-composition of cost $f(t) = t^{1/d}$ from the $\mathsf{NP}$-hard Multicolored Clique problem to Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph, where $t$ is the number of Multicolored Clique instances. We will construct an instance $(G', k', X')$ of Edge Dominating Set where all components of $G' - X'$ are isomorphic to $H$, implying that the result holds for all sets $\mathcal{H}$ containing $H$ (though a stronger lower bound may follow using another graph $H' \in \mathcal{H}$).

We choose the same equivalence relation $\mathcal{R}$ as in the proof of Theorem 9.1. Let a sequence of instances $I_i = (G_i, k)_{i=1}^{t}$ of Multicolored Clique be given that are in the same equivalence class of $\mathcal{R}$. As before, since all color classes have the same size we can identify for each color class the vertex sets. Let $V$ be the vertex set (of size $k \cdot n$) of the $t$ instances and let $V_1, V_2, \ldots, V_k$ be the different color classes (of size $n$). We assume, without loss of generality, that every instance has at least one edge in $E(V_p, V_q)$ for all $1 \leq p < q \leq k$. Otherwise, this instance would be a trivial no instance and we can delete it. We copy some instance until we have $\widetilde{t} = s^d$ instances, where $s$ is the least odd integer with $t \leq s^d$. It holds that $s = \lceil t^{1/d} \rceil$ or $s = \lceil t^{1/d} \rceil + 1$; hence

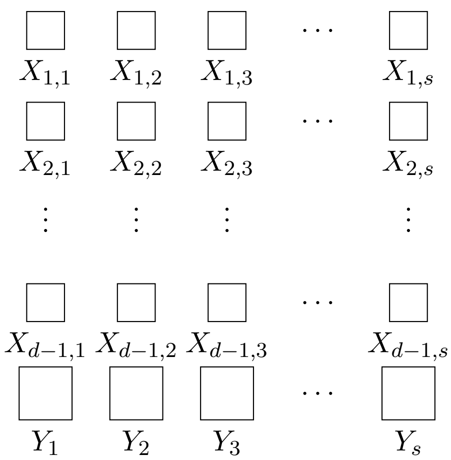



Figure 10.3.: The $d \cdot s$ sets that encode the $\widetilde{t} = s^d$ instances in the construction of $G'$.

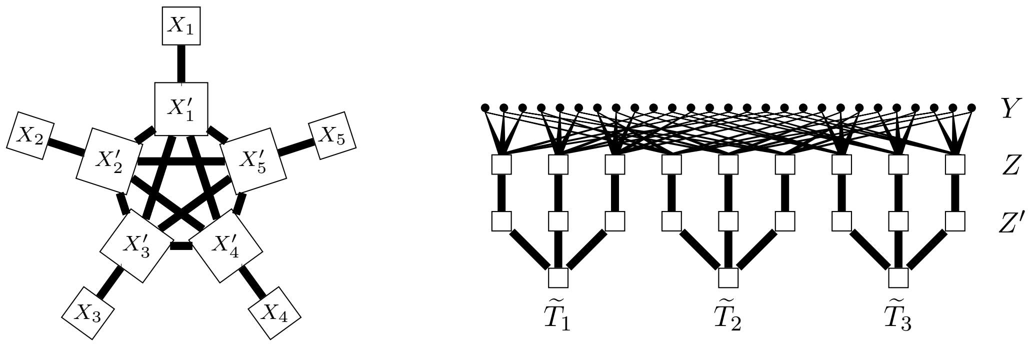
$s \leq t^{1/d} + 2$. Clearly, this does not affect whether at least one instance is a yes-instance for Multicolored Clique.

In the proof of Theorem 9.1 respectively Theorem 10.8 we add a set $W$ of size $2 \cdot \log(t)$ to the modulator to encode for each path $P_3$ respectively graph $H$ which instance it corresponds to. We cannot apply this construction here because we do not have the "control set" $C$ of a control pair; we only have a (large) strongly beneficial set $B$. Therefore, we have to find a different approach to encode to which instance a copy of the graph $H$ corresponds. Like Dell and Marx [DM12] we add $d \cdot s$ vertex sets to the graph $G'$ (more precisely the modulator $X'$), which form $d$ groups of size $s$ each. The goal is to associate each instance with a different choice of $d$ out of the $d \cdot s$ vertex sets, picking one from each group; there are $\widetilde{t} = s^d$ choices.

We construct an instance $(G', k', X')$ of Edge Dominating Set parameterized by the size of a modulator to an $H$-component graph; of course, this is also an instance of Edge Dominating Set parameterized by the size of modulator to an $\mathcal{H}$-component graph. (See Figures 10.3 and 10.4 for an illustration.)

We add $(d-1) \cdot s$ vertex sets, each of size $\binom{k}{2}$, to $G'$; we denote these sets by $X_{i,j}$ where $i \in [d-1]$ and $j \in [s]$. Every vertex in $X_{i,j}$, with $i \in [d-1]$ and $j \in [s]$, represents a different edge set $E(V_p, V_q)$ for $1 \leq p < q \leq k$. By $x_{i,j}^{p,q}$ we denote the vertex in $X_{i,j}$ that represents the edge set $E(V_p, V_q)$. Next, we add $s$ sets, each of size $\binom{k}{2} \cdot n^2$, to $G'$. We denote these sets by $Y_j$ with $j \in [s]$. Every vertex in $Y_j$, with $j \in [s]$, represents a possible edge (of a Multicolored Clique instance) between two vertices in different color classes $V_p$ and $V_q$, with $1 \leq p < q \leq k$. By $y_j^{\{u,v\}}$ we denote the vertex in $Y_j$ that represents the possible edge $\{u, v\}$ with $u \in V_p$, $v \in V_q$.

We modify the indexing of the input instances from using $i$ with $i \in [\widetilde{t}]$ to using index vectors $h = (h_1, h_2, \ldots, h_d) \in [s]^d$; there are $s^d = \widetilde{t}$ different index vectors. Henceforth, we refer to instances and their graphs through their index $h$. In the rest of the construc-




(a) selection gadget with $s = 5$ (b) clique gadget with $k = 3$ and $|V_i| = 3$

Figure 10.4.: Gadgets with notation as in the definition.

tion, every instance $(G_h, k)$ of MULTICOLORED CLIQUE with $h = (h_1, h_2, \dots, h_d) \in [s]^d$ only interacts with the vertex sets $X_{i,h_i}$ for $i \in [d-1]$ and the vertex set $Y_{h_d}$. For every instance $G_h$, with $h \in [s]^d$, we add $|E(G_h)|$ copies of the graph $H$ to $G'$ and denote the copy of $H$ that represents edge $e \in E(G_h)$ by $H_h^e$. Let $B_h^e = \{b_{h,1}^e, b_{h,2}^e, \dots, b_{h,d}^e\}$ be the copy of the beneficial set $B$ in $H_h^e$, i.e., vertex $b_{h,i}^e$ corresponds to vertex $b_i$ in $B$, for all $i \in [d]$. We add all edges $\{b_{h,i}^e, x_{i,h_i}^{p,q}\}$, with $i \in [d-1]$, as well as the edge $\{b_{h,d}^e, y_{h_d}^e\}$ to $G'$, with $h = (h_1, h_2, \dots, h_d) \in [s]^d$, $e \in E(G_h) \cap E(V_p, V_q)$ and $1 \le p < q \le k$. That is, an edge $e$ between color classes $V_p$ and $V_q$ in $G_h$ is (in part) represented by connecting its corresponding graph $H_h^e$ to the sets $X_{i,h_i}$ corresponding to $h$: The $i$th vertex $b_{h,i}^e$ of the beneficial set $B_h^e$ in $H_h^e$ is made adjacent to $x_{i,h_i}^{p,q} \in X_{i,h_i}$, for $i \in [d-1]$. These edges between $H_h^e$ and $X_{i,h_i}$ represent only the colors of the endpoints of $e$. Whereas, the edges between $H_h^e$ and $Y_{h_d}$ represent the endpoints of $e$: The vertex $b_{h,d}^e$ of the beneficial set $B_h^e$ in $H_h^e$ is made adjacent to $y_{h_d}^e$. Thus, every vertex $b_{h,i}^e$, with $i \in [d]$, is adjacent to exactly one vertex that is not contained in $V(H_h^e)$.

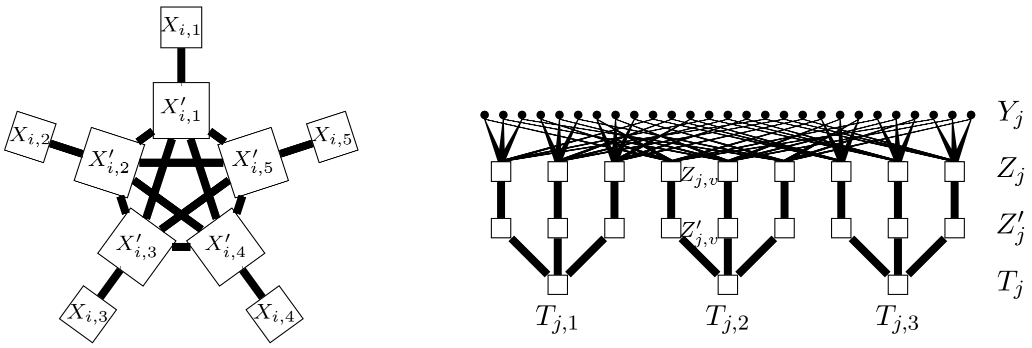
We need the sets $X_{i,j}$ with $i \in [d-1]$, and $j \in [s]$ only to encode the $\widetilde{t}$ instances; to make sure that there exists a clique in at least one instance we primarily use the sets $Y_j$ with $j \in [s]$. Our goal is that for every $i \in [d-1]$ exactly one of the sets $X_{i,1}, X_{i,2}, \dots X_{i,s}$ is contained in the set of endpoints of an edge dominating set of size at most $k'$ in $G'$. We obtain this by means of the following gadget:

A *selection gadget* of size $\alpha$ consists of $s$ sets, say $X_1', X_2', \dots, X_s'$, each of size $2 \cdot \binom{k}{2} \cdot d$ and $s$ sets, say $X_1, X_2, \dots, X_s$, each of size $\alpha$. Each vertex in $X_j'$, with $j \in [s]$, is connected to all vertices in $X_j$ and to all vertices in $X_{\bar{j}}'$ with $\bar{j} \in [s]$ and $\bar{j} \neq j$ (see Figure 10.4a for an illustration). Intuitively, with a local budget of $\frac{s-1}{2} \cdot 2 \cdot \binom{k}{2} \cdot d$ edges, one could cover exactly all $(s-1) \cdot 2 \cdot \binom{k}{2} \cdot d$ vertices of all but one set $X_j'$ by picking appropriate edges with endpoints in the other sets; here we use that $s$ is odd. This would force us to cover edges between $X_j'$ and $X_j$ by making all vertices in $X_j$ endpoints of solution edges with other endpoint outside of th selection gadget. We add $d-1$ selection gadgets of size $\binom{k}{2}$ to the modulator $X'$ and identify for each $j \in [s]$ the set $X_{i,j}$ with the set $X_j$ of one selection gadget. The $i$th selection gadget has the vertex sets $X_{i,1}, X_{i,2}, \ldots, X_{i,s}$ and $X'_{i,1}, X'_{i,2}, \ldots, X'_{i,s}$ where $i \in [d-1]$.

We still have to make sure that we pick $\binom{k}{2}$ edges that have their endpoints in a vertex set of size $k$, so that they must form a $k$-clique. To guarantee this, we add for each set $Y_j$, with $j \in [s]$, a gadget that we call clique gadget (see Figure 10.4b) to $G'$: A *clique gadget* consists of $k$ sets $\widetilde{T}_1, \widetilde{T}_2, \ldots, \widetilde{T}_k$ each of size $k-1$; every set represents one color class. Additionally, we add for every $v \in V$ a set $Z_v$ of size $k-1$ and a copy of $Z_v$, named $Z'_v$, to the gadget. The final set $Y$ of the gadget contains $\binom{k}{2} \cdot n^2$ vertices, one for each possible edge in an instance. (Each edge has its endpoints in different color classes and we have $k$ color classes of size $n$.) We denote the vertex in $Y$ that represents the edge between vertex $v$ and $u$ in different color classes by $y^{\{v,u\}}$. (Later we will identify $Y$ with one set $Y_j$ for $j \in [s]$.) We connect every vertex in $Z_v$ for a vertex $v \in V_i$, with $i \in [k]$, to all vertices in $Z'_v$, and to every vertex $y^{\{v,u\}}$ with $u \in V \setminus V_i$. Furthermore, we connect every vertex in $\widetilde{T}_i$, for $i \in [k]$, to all vertices in $Z'_v$, if $v \in V_i$.

This gadget is perhaps the most vital part of our construction (apart from understanding strongly beneficial sets in $H$-components). There are two different cases for its behavior, which we will trigger by another selection gadget. If there are no other constraints then it can be covered entirely by picking edges connecting sets $Z_v$ to sets $Z'_v$ (using $(k-1) \cdot k \cdot n$ edges). Else, as we will ensure for exactly one of these gadgets, the vertices in all sets $\widetilde{T}_i$ must be endpoints of solution edges because they have neighbors outside the gadget that are not contained in the solution. Nevertheless, we only want (have) $(k-1) \cdot k \cdot n$ solution edges that can be inside a clique gadget and we have to cover the vertices in all sets $\widetilde{T}_i$ only with this budget. To cover the vertices in $\widetilde{T}_i$ we add a matching between $\widetilde{T}_i$ and the vertices in one set $Z'_v$ for a single $v \in V_i$ to the solution. These $k$ vertices will be the vertices of a clique in one instance. Since, all sets $\widetilde{T}_i$, with $i \in [k]$, are covered, we can select $k-1$ edges each between the vertices of a set $Z_v$ (where $Z'_v$ is not covered by solution edges to $\widetilde{T}_1, \cup \ldots \cup \widetilde{T}_k$). We can pick these edges such that all, except the edges between $Z_v$ (where $v$ is a vertex in the "clique") and vertices in $y$ that represent an edge between $v$ and another vertex in the "clique", are dominated. We will dominate these $\binom{k}{2}$ edges in $Y$ that are incident with a non dominated edge via edges that have one endpoint in a copy of $H$, more precisely the vertex $b_d$ in a copy of $H$. This will guarantee that the edge between two clique vertices is an edge in the instance.

As mentioned above, we add $s$ clique gadgets to the modulator $X'$ and identify each $Y_j$, here $j \in [s]$, with a different set $Y$ of a clique gadget. To distinguish between sets in the different clique gadgets, we denote the other sets for the clique gadget containing set $Y_j$, with $j \in [s]$, by $T_{j,i}$, with $i \in [k]$, and by $Z_{j,v}$ and $Z'_{j,v}$, with $v \in V$; let $Z_j = \bigcup_{v \in V} Z_{j,v}$, let $Z'_j = \bigcup_{v \in V} Z'_{j,v}$ and let $T_j := \bigcup_{i=1}^{k} T_{j,i}$. Now, we have $s$ different clique gadgets and want to choose exactly one clique gadget where the set $T_j$ must be covered by the solution $F$. To this end, we add one last selection gadget of size $k \cdot (k-1)$ to $X'$ and identify the set $T_j$ with the set $X_j$ of the selection gadget, with $j \in [s]$. We denote the sets $X'_j$ of the selection gadget, with $j \in [s]$, by $T'_j$.




(a) the $i$th selection gadget with $s = 5$ (b) $j$th clique gadget with $k = 3$ and $|V_i| = 3$.

Figure 10.5.: Gadgets with notation as in the construction of $G'$.

The set $X'$ contains all vertices that are not contained in a copy of graph $H$; in total these are $(d-1) \cdot s \cdot \binom{k}{2} \cdot (1 + 2 \cdot d)$ vertices for the $d - 1$ selection gadgets of size $\binom{k}{2}$, $s \cdot ((k + 2 \cdot k \cdot n) \cdot (k-1) + \binom{k}{2} \cdot n^2)$ vertices for the $s$ clique gadgets and $s \cdot 2 \cdot \binom{k}{2} \cdot d$ vertices for the last selection gadget (only those that we did not already count, because they are also contained in a clique gadget); hence $|X'| \in \mathcal{O}(s \cdot n^2 \cdot k^2 \cdot d^2) \subseteq \mathcal{O}(t^{1/d} \cdot n^6)$, because $t^{1/d} \geq s - 2$ and $d, k \leq n$. Let

$$k' = d \cdot 2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2} + s \cdot k \cdot n \cdot (k-1) + \sum_{h \in [s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \binom{k}{2} \mathrm{cost}(B).$$

Intuitively, we have a local budget of $2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2}$ to dominate all edges of the complete $s$-partite graph that is contained in each of the $d$ selection gadgets, a local budget of $k \cdot n \cdot (k-1)$ to dominate all edges between $Z_j$ and $Z'_j$ in each of the $s$ clique gadgets, a local budget of $\sum_{h \in [s]^d} |E(G_h)| \cdot \textsc{eds}(H)$ to dominate all edges of $G - X$, and an extra budget of $\binom{k}{2}\mathrm{cost}(B)$ edges to dominate all remaining edges.

We have to show that there exists an index vector $h^* \in [s]^d$ such that $(G_{h^*}, k)$ is a yes-instance if and only if $(G', k', X')$ is a yes-instance.

($\Rightarrow$:) Assume that for some $h^* \in [s]^d$ the Multicolored Clique instance $(G_{h^*}, k)$ is a yes-instance. Let $X = \{x_1, x_2, \ldots, x_k\} \subseteq V$ be a multicolored clique of size $k$ in $G_{h^*}$ with $x_i \in V_i$ for $i \in [k]$ and let $E'$ be the set of edges of the clique $X$. Let $h^* = (h_1^*, h_2^*, \ldots, h_d^*) \in [s]^d$. We construct an edge dominating set $F$ of $G'$ as follows:

For each $i \in [d-1]$ we add a minimum edge dominating set in $G'[X'_{i,1} \cup X'_{i,2} \cup \ldots \cup X'_{i,s}]$ of size $2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2}$ to $F$ such that each set, except the set $X'_{i,h_i^*}$, is covered by $F$. Such a minimum edge dominating set exists, because $G'[X'_{i,1} \cup X'_{i,2} \cup \ldots \cup X'_{i,s}]$ is a complete $s$-partite graph and $s$ is odd. Thus, we dominate all edges, except the edges between the vertex sets $X'_{i,h_i^*}$ and $X_{i,h_i^*}$ in these $d-1$ selection gadgets.

Next, we add a minimum edge dominating set in $G'[T'_1 \cup T'_2 \cup \ldots \cup T'_s]$ of size $2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2}$ to $F$ such that each vertex set, except the set $T'_{h_d^*}$, is covered by $F$. (Such a minimum edge dominating set exists for the same reasons as above.) Consider the $s$ clique gadgets: For each $j \in \{1, 2, \ldots, s\} \setminus \{h_d^*\}$ we add a perfect matching between the vertex sets $Z_j$ and $Z'_j$ to $F$; such a matching of size $|V| \cdot (k-1) = n \cdot k \cdot (k-1)$ exists by construction (for each $v \in V$ it holds that every vertex in $Z_{j,v} \subseteq Z_j$ is connected to every vertex in $Z'_{j,v} \subseteq Z'_j$ and both sets have the same size). Thus, all edges in these clique gadgets and between these clique gadgets and the selection gadget are dominated: The only uncovered vertices in a clique gadget are the vertices $Y_j$ and $T_j$, with $j \in [s]$ and $j \neq h_d^*$. These sets are independent sets and only the set $T_j$ is adjacent to a selection gadget, more precisely, to the vertex set $T'_j$ which is covered by $F$.

Since the edges between the vertex sets $T'_{h_d^*}$ and $T_{h_d^*}$ are not dominated so far, we add a perfect matching between the vertex sets $T_{h_d^*}$ and $\{Z'_{h_d^*,x} \mid x \in X\}$ to $F$; such a matching of size $|X| \cdot (k-1) = k \cdot (k-1)$ exists by construction: the set $T_{h_d^*,i}$, with $i \in [k]$, has size $k-1$ and every vertex in $T_{h_d^*,i}$ is connected to all $k-1$ vertices in $Z'_{h_d^*,x_i}$ with $x_i \in X \cap V_i$. Thus, we covered all edges inside the selection gadget of size $k \cdot (k-1)$ and between this selection gadget and the clique gadgets. Next, we add for all $v \in V_i \setminus X$, with $i \in [k]$, a perfect matching between $Z_{h_d^*,v}$ and $\{y_{h_d^*}^{\{v,x\}} \mid x \in X - x_i\}$: Both sets have size $k-1$ and every vertex in $Z_{h_d^*,v}$ is adjacent to all vertices in $\{y_{h_d^*}^{\{v,x\}} \mid x \in X \setminus \{x_i\}\}$. In total, these are $|V \setminus X| \cdot (k-1) = (n-1) \cdot k \cdot (k-1)$ edges.

So far, we dominate all edges, except the edges between vertices in $Z_{h_d^*,x}$, with $x \in X$, and the vertices in $\{y_{h_d^*}^{\{x,y\}} \mid x, y \in X, x \neq y\}$: The sets $T_{h_d^*}$, $Z'_{h_d^*,x}$, with $x \in X$, and $Z_{h_d^*,v}$, with $v \in V \setminus X$, are covered by $F$. Thus, the only edges that are not dominated in this clique gadget are those between the vertex sets $Z_{h_d^*,x}$ with $x \in X$ and $Y_{h_d^*}$. A vertex in $Z_{h_d^*,x_i}$, with $i \in [k]$, is adjacent to a vertex $y_{h_d^*}^{\{u,v\}}$ if $u = x_i$ and $v \in V \setminus V_i$. But, for all $v \in V \setminus (V_i \cup X)$, the vertex $y_{h_d^*}^{\{x_i,v\}}$ is already covered by $F$ (see above).

Finally, we add an edge dominating set for the copies of $H$ to $F$. For all graphs $H_h^e$ with $h \in [s]^d$, and $e \in E(G_h)$, and either $h \neq h^*$ or $e \notin E'$ we add a minimum edge dominating set in $H_h^e$ that covers all vertices in $B_h^e$ to $F$; such a minimum edge dominating set exists by assumption. (Recall, $E'$ is the set of edges between vertices in $X$.) For all graphs $H_{h^*}^e$ with $e = \{x_p, x_q\} \in E'$, with $1 \leq p < q \leq k$, we add a minimum edge dominating set in $H_{h^*}^e - B_{h^*}^e$ to $F$ as well as the edges $\{b_{h^*,i}^e, x_{i,h_i^*}^{p,q}\}$, with $i \in [d-1]$, and the edge $\{b_{h^*,d}^e, y_{h_d^*}^e\}$. These edges exist by construction, because $E' \subseteq E(G_{h^*})$. Thus, the set $V(F)$ contains the vertex set $X_{i,h_i^*}$, with $i \in [d-1]$, and the vertex set $\{y_{h_d^*}^{\{x,y\}} \mid x, y \in X, x \neq y\}$, which implies that $F$ dominates all edges that are contained in a clique gadget and in a selection gadget.

Since all vertices in $B_h^i$, with $h \in [s]^d$ and $i \in [k]$, are dominated by $F$ and these are the only vertices in the connected component of $H_h^e$ that are adjacent to a vertex in $X'$, and $F$ dominates all clique gadgets, selection gadgets, and connected components of $G' - X'$, the set $F$ is an edge dominating set of $G'$.

The set $F$ contains $d \cdot 2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2}$ edges inside the selection gadgets, $(s-1) \cdot n \cdot k \cdot (k-1)$ edges inside the clique gadgets that do not contain $Y_{h^*_d}$, $k \cdot (k-1) + (n-1) \cdot k \cdot (k-1)$ edges inside the clique gadget that contains $Y_{h^*_d}$, $\textsc{eds}(H)$ edges for all graphs $H^e_h$ with $h \in [s]^d$, $e \in E(G_h)$ and either $h \neq h^*$ or $e \notin E'$, and $\textsc{eds}(H - B) + |B| = \textsc{eds}(H) + \text{cost}(B)$ edges for all graphs $H^e_{h^*}$ with $e \in E'$. This sums up to $k'$, implying that $(G', k', X')$ is a yes-instance.

($\Leftarrow$:) Assume that $(G', k', X')$ is a yes-instance of Edge Dominating Set and let $F$ be an edge dominating set of size at most $k'$ in $G'$. First, we consider how the edge dominating set $F$ interacts with the graph $G'$:

- We need at least $k \cdot n \cdot (k-1)$ edges to dominate all edges between $Z_j = \bigcup_{v \in V} Z_{j,v}$, and $Z'_j = \bigcup_{v \in V} Z'_{j,v}$ in one clique gadget, because $G'[Z_{j,v} \cup Z'_{j,v}]$ is a complete bipartite graph whose bipartition has the parts $Z_{j,v}$ and $Z'_{j,v}$ for all $v \in V$ and $|Z_{j,v}| = |Z'_{j,v}| = k - 1$. Thus, at least $k \cdot n \cdot (k-1)$ edges of $F$ must be contained inside a clique gadget because we need at least $k \cdot n \cdot (k-1)$ edges to dominate all edges between $Z_j$ and $Z'_j$, with $j \in [s]$, and because these sets are only adjacent to vertices inside the clique gadget they belong to.
- Furthermore, $F$ contains at least $2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2}$ edges inside each selection gadget, because the $s$ sets $X'_{i,1}, X'_{i,2}, \ldots, X'_{i,s}$, with $i \in [d-1]$, respectively the $s$ sets $T'_1, T'_2, \ldots, T'_s$ of each selection gadget form a complete $s$-partite graph where each partition has size $2 \cdot \binom{k}{2} \cdot d$ and these sets are only adjacent to vertices in their selection gadget. Note that, the sets $T_1, T_2, \ldots, T_s$ are contained in one selection gadget and in the clique gadgets; but our counting is still correct, because each of the $k \cdot n \cdot (k-1)$ edges that are contained in the clique gadgets must have at least one endpoint in the vertex set $Z_j \cup Z'_j$, with $j \in [s]$, and each of the $2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2}$ edges in the selection gadgets must have at least one endpoint in the vertex set $T'_1 \cup T'_2 \cup \ldots \cup T'_s$, and because the sets $Z_j \cup Z'_j$ and $T'_1 \cup T'_2 \cup \ldots \cup T'_s$ are not adjacent.
- To dominate all edges in $H^e_h$, with $h \in [s]^d$ and $e \in E(G_h)$, we need at least $\textsc{eds}(H)$ edges that are adjacent to $V(H^e_h)$. Thus, we need at least $\sum_{h \in [s]^d} |E(G_h)| \cdot \textsc{eds}(H)$ edges to dominate all edges of $G' - X'$.

Summarizing, for all, except $\binom{k}{2} \cdot \text{cost}(B)$ edges of $F$ we know at least one endpoint and that these edges are either contained in a selection gadget, a clique gadget, or adjacent to a copy of $H$. During the proof, we will show that we can make some assumptions about the edge dominating set $F$. To achieve these assumptions, we replace some edges in $F$ such that the resulting graph is still an edge dominating set of size $|F|$ in $G'$. But, a replacement of an edge will always preserve the previous assumptions.

Claim 10.24. *There exists an edge dominating set $F'$ of size $k'$ in $G'$ such that for each $i \in [d-1]$ there exists exactly one $j \in [s]$ such that no vertex in $X'_{i,j}$ is covered by $F'$ (hence $X'_{i,j} \cap V(F') = \emptyset$), and there exists exactly one $j \in [s]$ such that no vertex in $T'_j$ is covered by $F'$ (hence $T'_j \cap V(F') = \emptyset$).*

Proof. Since the $s$ sets $X'_{i,1}, X'_{i,2}, \ldots, X'_{i,s}$, with $i \in [d-1]$, respectively the $s$ sets $T'_1, T'_2, \ldots, T'_s$ of each selection gadget form a complete $s$-partite graph, it holds that $V(F)$ contains at least $s-1$ sets of the $s$ sets $X'_{i,1}, X'_{i,2}, \ldots, X'_{i,s}$, with $i \in [d-1]$, respectively at least $s-1$ of the $s$ sets $T'_1, T'_2, \ldots, T'_s$. First, we show that not all sets $X'_{i,1}, X'_{i,2}, \ldots, X'_{i,s}$, with $i \in [d-1]$, respectively not all sets $T'_1, T'_2, \ldots, T'_s$ can be covered by $F$:

Assume that all vertices in $X'_{i,1} \cup X'_{i,2} \cup \ldots \cup X'_{i,s}$, for some $i \in [d-1]$, respectively all vertices in $T'_1 \cup T'_2 \cup \ldots \cup T'_s$ are contained in $V(F)$. This would imply that at least

$$\frac{1}{2} \cdot \left| \bigcup_{j=1}^{s} X'_{i,j} \right| = \frac{1}{2} \cdot \left| \bigcup_{j=1}^{s} T'_j \right| = \frac{1}{2} \cdot s \cdot 2 \cdot \binom{k}{2} \cdot d = 2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2} + \binom{k}{2} \cdot d$$

edges of $F$ must be contained in this selection gadget. These are at least $\binom{k}{2} \cdot d$ edges more than the minimum number of edges in $F$ that must be contained in a selection gadget. But, we showed above that $F$ has at most $\binom{k}{2} \cdot \mathrm{cost}(B) < \binom{k}{2} \cdot |B| = \binom{k}{2} \cdot d$ additional edges. Thus, if $V(F)$ contains the entire set $X'_{i,1} \cup X'_{i,2} \cup \ldots \cup X'_{i,s}$, for some $i \in [d-1]$, respectively the entire set $T'_1 \cup T'_2 \cup \ldots \cup T'_s$, then $F$ contains more than $k'$ edges which is a contradiction.

Thus, for each $i \in [d-1]$ there exists an $h_i \in [s]$ such that not all vertices in $X'_{i,h_i}$ are covered by $F$ and there exists an $h_d \in [s]$ such that not all vertices in $T'_{h_d}$ are covered by $F$. Let $h = (h_1, h_2, \ldots, h_d)$. Since all vertices in $X'_{i,h_i}$, with $i \in [d-1]$, respectively all vertices in $T'_{h_d}$ have the same neighborhood and at least one vertex in these sets in not contained in $V(F)$, it holds that all vertices in the neighborhood of $X'_{i,h_i}$, with $i \in [d-1]$, respectively in the neighborhood of $T'_{h_d}$ must be contained in $V(F)$; otherwise $F$ would not be an edge dominating set in $G'$.

We replace every edge in $F$ that is incident with a vertex in $X'_{i,h_i}$, with $i \in [d-1]$, respectively to a vertex in $T'_{h_d}$. (Note, that the set $\bigcup_{i=1}^{d-1} X'_{i,h_i} \cup T'_{h_d}$ is an independent set, thus every edge in $F$ that is incident with this set must have its other endpoint outside this set.) Let $f = \{u, v\} \in F$ be an edge in $F$ with $u \in \bigcup_{i=1}^{d-1} X'_{i,h_i} \cup T'_{h_d}$. By construction, the vertex $v$ is a vertex in $X'_{i,j}$, with $i \in [d-1]$, $j \in [s]$ and $j \neq h_i$, or $T'_j$, with $j \in [s]$ and $j \neq h_d$, or $X_{i,h_i}$, with $i \in [d-1]$, or $T_{h_d}$. Hence, $v$ has a neighbor $v'$ that is not contained in $\bigcup_{i=1}^{d-1} X'_{i,h_i} \cup T'_{h_d}$. Thus, we can replace edge $f$ in $F$ with edge $f' = \{v, v'\}$ to obtain $F'$. The set $F'$ is still an edge dominating set: the only vertices that are not covered by the set $F'$ any more are contained in $\bigcup_{i=1}^{d-1} X'_{i,h_i} \cup T'_{h_d}$, but this set is an independent set and the neighborhood of this set is still covered by $F'$. This proves the claim. □

Assume that the edge dominating set $F$ fulfills the properties of Claim 10.24 (if this is not the case we can replace $F$ by $F'$). Let $h^* = (h^*_1, h^*_2, \ldots, h^*_d) \in [s]^d$ such that no vertex in $X'_{i,h^*_i}$, for $i \in [d-1]$, is covered by $F$ and no vertex in $T'_{h^*_d}$ is covered by $F$. It follows, that the sets $X_{i,h^*_i}$, for $i \in [d-1]$, must be covered by $F$ because all vertices in $X'_{i,h^*_i}$ are adjacent to all vertices in $X_{i,h^*_i}$. The vertex sets $X_{i,h^*_i}$, with $i \in [d-1]$, are only adjacent to the sets $X'_{i,h_i^*}$ and copies of $H$; hence the edges of $F$ that cover $X_{i,h_i^*}$ have their other endpoint in a copy of $H$.

Let $F_{h^*} = \{f \in F \mid \exists i \in [d-1]\colon f \cap X_{i,h_i^*} \neq \emptyset\}$ be the set of edges in $F$ that are incident with a vertex in $X_{i,h_i^*}$, with $i \in [d-1]$, and let $F_h^e = \{f \in F \mid f \cap V(H_h^e) \neq \emptyset\}$ be the set of edges in $F$ that are incident with a vertex in $H_h^e$ with $h \in [s]^d$. Let $B_h^e(F_{h^*}) = \{b \in B_h^e \mid \exists f \in F_{h^*}\colon b \in f\}$ be the set of vertices in $B_h^e$ that are incident with an edge in $F_{h^*}$. It holds that $F_h^e$, with $h \in [s]^d$ and $e \in E(G_h)$, has at least the size of a minimum edge dominating set in $H_h^e - B_h^e(F_{h^*})$ plus the size of $B_h^e(F_{h^*})$, because the edges in $F_h^e$ that have one endpoint in $B_h^e(F_{h^*})$ have their other endpoint not in $H_h^e$ and to dominate all remaining edges in $H_h^e$ we need at least $\textsc{eds}(H_h^e - B_h^e(F_{h^*}))$ many edges. Since no two copies of $H$ are adjacent, this implies that at least

$$\sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*}))$$

edges of $F$ are incident with a copy of $H$ because

$$\begin{aligned}\sum_{h\in[s]^d} \sum_{e\in E(G_h)} |F_h^e| &\geq \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \left(\textsc{eds}(H - B_h^e(F_{h^*})) + |B_h^e(F_{h^*})|\right)\\ &= \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \left(\textsc{eds}(H) + \mathrm{cost}(B_h^e(F_{h^*}))\right)\\ &= \sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*})). \end{aligned} \tag{10.2}$$

Now, we have $\sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*}))$ edges more in $F$ that are incident with a copy of $H$ than the lower bound of $\sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H)$ edges. These edges belong neither to the $2 \cdot \binom{k}{2} \cdot d \cdot \frac{s-1}{2}$ edges that we need to dominate all edges in the complete $s$-partite graph that is a subgraph of every selection gadget nor to the $k \cdot n \cdot (k-1)$ edges that we need to dominate one clique gadget. Thus,

$$\sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*})) \leq \binom{k}{2} \cdot \mathrm{cost}(B).$$

Claim 10.25.

$$\sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*})) = \binom{k}{2} \cdot \mathrm{cost}(B).$$

Proof. First, we rewrite the left-hand side as follows:

$$\sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*})) = \sum_{1\leq p<q\leq k} \sum_{h\in[s]^d} \sum_{e\in E(G_h)\cap E(V_p,V_q)} \mathrm{cost}(B_h^e(F_{h^*})).$$

We assume for the sake of contradiction that

$$\sum_{1\leq p<q\leq k}\sum_{h\in[s]^d}\sum_{e\in E(G_h)\cap E(V_p,V_q)}\mathrm{cost}(B_h^e(F_{h^*}))<\binom{k}{2}\cdot\mathrm{cost}(B).$$

This implies that there exist $1\leq\bar{p}<\bar{q}\leq k$ such that

$$\sum_{h\in[s]^d}\sum_{e\in E(G_h)\cap E(V_{\bar{p}},V_{\bar{q}})}\mathrm{cost}(B_h^e(F_{h^*}))<\mathrm{cost}(B).$$

We will show that $B':=\bigcup_{h\in[s]^d}\bigcup_{e\in E(G_h)\cap E(V_{\bar{p}},V_{\bar{q}})}B_h^e(F_{h^*})$ contains at least one copy of every vertex in $B\setminus\{b_d\}$. Let $i\in[d-1]$. Consider vertex $x_{i,h_i^*}^{\bar{p},\bar{q}}$ in $X_{i,h_i^*}$, which is covered by an edge in $F_{h^*}$ (by definition of $F_{h^*}$) and let $f=\{x_{i,h_i^*}^{\bar{p},\bar{q}},v\}$ be an edge in $F_{h^*}$ that has $x_{i,h_i^*}^{\bar{p},\bar{q}}$ as one endpoint. The vertex $v$ must be a vertex in a copy of $H$ because $X_{i,h_i^*}$ is only adjacent to vertices in $X'_{i,h_i^*}$ (which are not covered by $F$), and adjacent to copies of $H$. More precisely, since $f$ is an edge in $E(G')$, the vertex $v$ must be contained in $\{b_{h,i}^e\mid h\in[s]^d,h_i=h_i^*,e\in E(G_h)\cap E(V_{\bar{p}},V_{\bar{q}})\}$ (construction of $G'$: we add edges $\{x_{i,h_i}^e,b_{h,i}^e\}$ to $G'$ with $h=(h_1,h_2,\ldots,h_d)\in[s]^d$ and $e\in E(G_h)$). But, every vertex in this set is a copy of $b_i$ and it follows that $B'$ contains at least one copy of $b_i$. Since, this holds for all $i\in[d-1]$ it follows that $B'$ contains at least one copy of each vertex in $B\setminus\{b_d\}$.

Let $B_1,B_2,\ldots,B_l$ be the subsets of $B$ that correspond to the non-empty sets in $\{B_h^e(F_{h^*})\mid h\in[s]^d,e\in E(G_h)\cap E(V_{\bar{p}},V_{\bar{q}})\}$. Now, the sets $B_1,B_2,\ldots,B_l$ together with the set $\{b_d\}$ cover the set $B$. Since the vertex $b_d$ is not extendable in $H$ (Proposition 10.6 item (8)) it holds that $\mathrm{cost}(\{b_d\})=1$. Thus, it holds that $\sum_{i=1}^l\mathrm{cost}(B_i)+\mathrm{cost}(\{b_d\})<\mathrm{cost}(B)+1$, because we assumed that $\sum_{i=1}^l\mathrm{cost}(B_i)<\mathrm{cost}(B)$ and $\mathrm{cost}(\{b_d\})=1$; hence $\sum_{i=1}^l\mathrm{cost}(B_i)+\mathrm{cost}(\{b_d\})\leq\mathrm{cost}(B)$. Note that every set $B_i$, with $i\in[l]$, must be a proper subset of $B$; otherwise $\mathrm{cost}(B)=\mathrm{cost}(B_i)$ which contradicts the assumption that $\sum_{i=1}^l\mathrm{cost}(B_i)<\mathrm{cost}(B)$. Summarized, the sets $B_1,B_2,\ldots,B_l,\{b_d\}\subsetneq B$ cover $B$ and it holds that $\sum_{i=1}^l\mathrm{cost}(B_i)+\mathrm{cost}(\{b_d\})\leq\mathrm{cost}(B)$. This implies that $B$ is not strongly beneficial (see definition), which is a contradiction and proves the claim. □

So far, we know that $F$ contains $d\cdot 2\cdot\binom{k}{2}\cdot d\cdot\frac{s-1}{2}$ edges that cover the $d$ different complete $s$-partite graphs that are subgraphs of the $d$ selection gadgets, and that (at least) $\sum_{h\in[s]^d}|E(G_h)|\cdot\textsc{eds}(H)+\binom{k}{2}\mathrm{cost}(B)$ edges are incident with copies of $H$ (Claim 10.25). Thus, the remaining $s\cdot k\cdot n\cdot(k-1)$ edges must cover the $s$ clique gadgets. Since this is the number of edges we need (at least) to dominate the edges between the vertex sets $Z_j$ and $Z'_j$, with $j\in[s]$, in a clique gadget, every remaining edge must either be incident with a vertex of $Z_j$ or with a vertex of $Z'_j$.

Consider the clique gadget that contains the vertex set $Y_{h_d^*}$. Since no vertex in $T'_{h_d^*}$ is covered by $F$, it holds that every vertex in $T_{h_d^*}$ must be covered by $F$ (because every vertex in $T'_{h_d^*}$ is adjacent to every vertex in $T_{h_d^*}$). Each vertex in $T_{h_d^*}$ is only adjacent to vertices in $T'_{h_d^*}$ and $Z'_{h_d^*}$; thus, every edge in $F$ that is incident with a vertex in $T_{h_d^*}$ has its other endpoint in $Z'_{h_d^*}$. Furthermore, for each vertex $v \in V$ the entire set $Z_{h_d^*,v}$ or the entire set $Z'_{h_d^*,v}$ is contained in $V(F)$ (both is also okay) because $G'[Z_{h_d^*,v} \cup Z'_{h_d^*,v}]$ is a complete bipartite graph whose partition has the parts $Z_{h_d^*,v}$ and $Z'_{h_d^*,v}$. Additionally, this implies that the vertex set $Z_{h_d^*,v} \cup Z'_{h_d^*,v}$ is incident with exactly $|Z_{h_d^*,v}| = |Z'_{h_d^*,v}| = k-1$ edges of $F$ because $|V| = k \cdot n$, and there are $k \cdot n \cdot (k-1)$ edges in $F$ that dominate all edges between $Z_{h_d^*}$ and $Z'_{h_d^*}$.

Claim 10.26. *There exists an edge dominating set $F'$ of size $k'$ in $G'$ such that for all vertices $v \in V$ either no vertex in $Z'_{h_d^*,v}$ or no vertex in $Z_{h_d^*,v}$ is covered by $F'$. Furthermore, for each color class $V_i$, with $i \in [k]$, there exists exactly one vertex $v_i \in V_i$ such that $F'$ covers no vertex in $Z_{h_d^*,v}$.*

Proof. Let $i \in [k]$ and let $v_i$ be a vertex in $V_i$ such that $F$ contains an edge that has one endpoint in $T_{h_d^*,i}$ and the other endpoint in $Z'_{h_d^*,v_i}$. Since the set $Z'_{h_d^*,v_i} \cup Z_{h_d^*,v_i}$ is only incident with $k-1$ edges of $F$, and since either $Z'_{h_d^*,v_i}$ or $Z_{h_d^*,v_i}$ must be entirely contained in $V(F)$ it holds that $Z'_{h_d^*,v_i}$ is entirely covered by $F$, and that every edge of $F$ that is incident with a vertex in $Z_{h_d^*,v_i}$ has its other endpoint in $Z'_{h_d^*,v_i}$. Furthermore, there exists a vertex in $Z_{h_d^*,v_i}$ that is not covered by $F$ because the set $Z'_{h_d^*,v_i} \cup Z_{h_d^*,v_i}$ is only incident with $k-1$ edges of $F$ and at least one of these $k-1$ edge has no endpoint in $Z_{h_d^*,v_i}$. Thus, the entire neighborhood of $Z_{h_d^*,v_i}$ must be covered by $F$ because all vertices in $Z_{h_d^*,v_i}$ have the same neighborhood. Hence, we can delete every edge in $F$ that has one endpoint in $Z'_{h_d^*,v_i}$ and add a maximum matching of the complete bipartite graph $G'[Z'_{h_d^*,v_i} \cup T_{h_d^*,i}]$ to $F$. This maximum matching has size $|Z'_{h_d^*,v_i}| = |T_{h_d^*,i}|$ and covers all vertices in $Z'_{h_d^*,v_i} \cup T_{h_d^*,i}$. The resulting edge set, which we denote by $\widetilde{F}$ is still an edge dominating set of size $|F|$ in $G'$ because the only vertices that are not covered anymore are contained in $Z_{h_d^*,v_i}$, but all neighbors are still covered: The vertices in $Z'_{h_d^*,v_i}$ are only adjacent to the vertices in $Z_{h_d^*,v_i}$ and $T_{h_d^*,i}$, and the vertices in $Z_{h_d^*,v_i}$ are adjacent to vertices in $Z'_{h_d^*,v_i}$ and $Y_{h_d^*}$. Thus, the only edges that we replace and that are incident with $Z_{h_d^*,v_i}$ or a neighbor of $Z_{h_d^*,v_i}$ are incident with $Z'_{h_d^*,v_i}$ and these vertices are still covered by $\widetilde{F}$. It holds that the edge dominating set $\widetilde{F}$ covers all vertices in $Z'_{h_d^*,v_i}$ and no vertex in $Z_{h_d^*,v_i}$.

Now, consider a vertex $v \in V_i \setminus \{v_i\}$. Every vertex in $Z'_{h_d^*,v}$ is only adjacent to the vertices in $T_{h_d^*,i} \cup Z_{h_d^*,v}$. Since every vertex in $T_{h_d^*,i}$ is covered by $\widetilde{F}$, we can replace the edges in $\widetilde{F}$ that are incident with a vertex in $Z_{h_d^*,v}$. Recall that either the set vertex $Z'_{h_d^*,v}$ or the vertex set $Z_{h_d^*,v}$ is entirely covered by $\widetilde{F}$, and that $Z'_{h_d^*,v} \cup Z_{h_d^*,v}$ is only incident with $|Z'_{h_d^*,v}| = |Z_{h_d^*,v}| = k-1$ edges of $\widetilde{F}$. If $\widetilde{F}$ covers all vertices in $Z'_{h_d^*,v}$ then we replace every edge $e = \{z, z'\} \in F$ with $z' \in Z'_{h_d^*,v}$ and $z \in Z_{h_d^*,v}$ by an edge in $E(z, Y_{h_d^*})$. Otherwise, if $\widetilde{F}$ covers all vertices in $Z_{h_d^*,v}$ then we delete the

$|Z'_{h^*_d,v}| = |Z_{h^*_d,v}| = k-1$ edges in $F$ that are incident with a vertex in $Z'_{h^*_d,v} \cup Z_{h^*_d,v}$ and add for each vertex $z$ in $Z_{h^*_d,v}$ exactly one edge in $E(z, Y_{h^*_d})$ to $\widetilde{F}$. We denote the resulting set by $F'$. Note that in both cases the set $Z'_{h^*_d,v}$ is not covered by $F'$ and the set $Z_{h^*_d,v}$ is covered by $F'$. Clearly, $F'$ has the same size as $\widetilde{F}$ (by construction) and hence as $F$. The fact that $F'$ is still an edge dominating set holds because the only vertices that are possibly covered by $\widetilde{F}$ and not by $F'$ are contained in $Z'_{h^*_d,v}$, but all neighbors of $Z'_{h^*_d,v}$ are still contained in $V(F')$. We can do this for all $i \in [k]$ and all $v \in V_i \setminus \{v_i\}$ independently; this proves the claim. $\square$

Let $F$ be the edge dominating set that we construct during the proof of Claim 10.26. Now, for each $i \in [k]$ there exists exactly one vertex $v$ in $V_i$ such that no vertex in $Z_{h^*_d,v}$ is incident with an edge in $F$; denote this vertex by $x_i$. Let $X = \{x_1, x_2, \ldots, x_k\}$. We will show that $X$ is a clique in $G_{h^*}$.

Every vertex in the set $Z_{h^*_d,x_i}$, with $i \in [k]$, is adjacent to all vertices in $Z'_{h^*_d,x_i}$ and to all vertices in $\{y^{\{x_i,v\}}_{h^*_d} \in Y_{h^*_d} \mid v \in V \setminus V_i\}$; thus both sets must be covered by $F$. The second set contains $(k-1) \cdot n$ vertices, one vertex for every vertex in $V \setminus V_i$. All vertices $y^{\{x_i,x_j\}}_{h^*_d}$, with $j \in [k]$ and $j \neq i$, can only be covered by edges in $F$ that have one endpoint in a copy of $H$ because these vertices are only adjacent to copies of $H$, and adjacent to the set $Z_{h^*_d,x_i} \cup Z_{h^*_d,x_j}$, which is not incident with an edge in $F$. Let $Y = \{y^{\{x_i,x_j\}}_{h^*_d} \in Y_{h^*_d} \mid 1 \leq i < j \leq k\}$ be the set of vertices in $Y_{h^*_d}$ that must be covered by $F$ via edges that have one endpoint in a copy of $H$. This set has size $\binom{k}{2}$.

Recall that $F^e_h$ is the set of edges in $F$ that are incident with a vertex in $H^e_h$. The sets $F^e_h$ are pairwise disjoint because no two copies of $H$ are adjacent. Let $B^e_h(F) = \{b \in B^e_h \mid \exists f \in F\colon b \in f \text{ and } f \nsubseteq V(H^e_h)\}$ be the set of vertices in $B^e_h$ that are an endpoint of an edge $f$ in $F$ whose other endpoint is not a vertex in a copy of $H$; thus, by construction, this other endpoint is contained in a set $X_{i,j}$ or a set $Y_j$, with $i \in [d-1]$ and $j \in [s]$. Note that $B^e_h(F_{h^*}) \subseteq B^e_h(F)$ because $B^e_h(F_{h^*})$ contains all edges that have one endpoint in $H^e_h$ (or more precisely $B^e_h$) and the other endpoint in a set $X_{i,h^*_i}$ ,with $i \in [d-1]$, whereas, $B^e_h(F_h)$ contains all edges that have one endpoint in $H^e_h$ (or more precisely $B^e_h$) and the other endpoint in $X'$. As before, the edge set $F^e_h$, with $h \in [s]^d$ and $e \in E(G_h)$, has at least the size of a minimum edge dominating set in $H^e_h - B^e_h(F)$ plus the size of $B^e_h(F)$ because $F^e_h$ contains the $|B^e_h(F)|$ edges between the vertices in $B^e_h(F)$ and a vertex that is not in $V(H^e_h)$, and because $F^e_h$ must also dominate all remaining edges in $H^e_h - B^e_h(F)$. Thus, it holds:

$$\begin{aligned}\sum_{h\in[s]^d}\sum_{e\in E(G_h)} |F^e_h| &\geq \sum_{h\in[s]^d}\sum_{e\in E(G_h)} \left(\textsc{eds}(H^e_h - B^e_h(F)) + |B^e_h(F)|\right) \\ &= \sum_{h\in[s]^d}\sum_{e\in E(G_h)} \left(\textsc{eds}(H) + \operatorname{cost}(B^e_h(F))\right). \end{aligned} \tag{10.3}$$

Since every edge in a set $F_h^e$, for $h \in [s]^d$ and $e \in E(G_h)$, is incident with a vertex in $H_h^e$ (and therefore neither incident with a vertex in $Z_j \cup Z_j'$, for $j \in [s]$, nor incident with a vertex in a selection gadget) we know that

$$\sum_{h\in[s]^d} \sum_{e\in E(G_h)} |F_h^e| \leq \sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \binom{k}{2} \cdot \mathrm{cost}(B).$$

It follows from Claim 10.25 together with inequality (10.2) that

$$\sum_{h\in[s]^d} \sum_{e\in E(G_h)} |F_h^e| \geq \sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \binom{k}{2} \cdot \mathrm{cost}(B).$$

Combining the two last inequalities we obtain that

$$\sum_{h\in[s]^d} \sum_{e\in E(G_h)} |F_h^e| = \sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \binom{k}{2} \cdot \mathrm{cost}(B).$$

Summarized, this implies:

$$\begin{aligned}
&\sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \binom{k}{2} \cdot \mathrm{cost}(B) \\
&\quad\overset{(10.4)}{=} \sum_{h\in[s]^d} \sum_{e\in E(G_h)} |F_h^e| \\
&\quad\overset{(10.3)}{\geq} \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \left(\textsc{eds}(H) + \mathrm{cost}(B_h^e(F))\right) \\
&\quad= \sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F)) \\
&\quad\geq \sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*})) \\
&\quad\overset{\textit{Claim } 10.25}{=} \sum_{h\in[s]^d} |E(G_h)| \cdot \textsc{eds}(H) + \binom{k}{2} \cdot \mathrm{cost}(B)
\end{aligned}$$

The last inequality holds because $B_h^e(F_{h^*}) \subseteq B_h^e(F)$, which implies that $\mathrm{cost}(B_h^e(F_{h^*})) \leq \mathrm{cost}(B_h^e(F))$ (Proposition 10.6 (5)). It follows that all terms are equal and, hence,

$$\begin{aligned}
\sum_{1\leq p<q\leq k} \sum_{h\in[s]^d} \sum_{e\in E(G_h)\cap E(V_p,V_q)} \mathrm{cost}(B_h^e(F)) &= \sum_{h\in[s]^d} \sum_{e\in E(G_h)} \mathrm{cost}(B_h^e(F_{h^*})) \\
&= \binom{k}{2} \cdot \mathrm{cost}(B).
\end{aligned}$$

This implies that either for all $1 \leq p < q \leq k$ it holds that

$$\sum_{h \in [s]^d} \sum_{e \in E(G_h) \cap E(V_p, V_q)} \text{cost}(B_h^e(F)) = \text{cost}(B),$$

or that there exist $1 \leq \bar{p} < \bar{q} \leq k$ such that

$$\sum_{h \in [s]^d} \sum_{e \in E(G_h) \cap E(V_{\bar{p}}, V_{\bar{q}})} \text{cost}(B_h^e(F)) < \text{cost}(B).$$

We will show that for all $1 \leq p < q \leq k$ it holds that

$$\sum_{h \in [s]^d} \sum_{e \in E(G_h) \cap E(V_p, V_q)} \text{cost}(B_h^e(F)) = \text{cost}(B), \tag{10.4}$$

and that $B_h^e(F) = B_h^e$ if and only if $h = h^*$ and edge $e$ has both endpoints in the set $X = \{x_1, x_2, \ldots, x_k\}$.

First, as in the proof of Theorem 10.8 we will show that we always have equality, hence that (10.4) holds. Let $1 \leq \bar{p} < \bar{q} \leq k$. We showed in the proof of Claim 10.25 that $B' := \bigcup_{h \in [s]^d} \bigcup_{e \in E(G_h) \cap E(V_{\bar{p}}, V_{\bar{q}})} B_h^e(F_{h^*})$ contains at least one copy of every vertex in $B \setminus \{b_d\}$. Thus, $B'' := \bigcup_{h \in [s]^d} \bigcup_{e \in E(G_h) \cap E(V_{\bar{p}}, V_{\bar{q}})} B_h^e(F)$ contains at least one copy of every vertex in $B \setminus \{b_d\}$, because $B_h^e(F_{h^*}) \subseteq B_h^e(F)$. Furthermore, $B''$ also contains a copy of $b_d$, because vertex $y_{h_d^*}^{\{x_{\bar{p}}, x_{\bar{q}}\}}$ must be covered by an edge $f$ in $F$ that has its other endpoint in a copy of $H$. By construction, the vertices in a copy of $H$ that are adjacent to vertex $y_{h_d^*}^{\{x_{\bar{p}}, x_{\bar{q}}\}}$ are the vertices $b_{h,d}^{\{x_{\bar{p}}, x_{\bar{q}}\}}$ with $h \in [s]^d$ s.t. $h_d = h_d^*$, and $\{x_{\bar{p}}, x_{\bar{q}}\} \in E(G_h) \cap E(V_{\bar{p}}, V_{\bar{q}})$; this implies that $B''$ contains a copy of $b_d$.

Now, let $B_1, B_2, \ldots, B_l$ be the subsets of $B$ that correspond to non-empty sets in $\{B_h^e(F) \mid h \in [s]^d, e \in E(G_h) \cap E(V_{\bar{p}}, V_{\bar{q}})\}$. Note that every set $B_i$ is either a proper subset of $B$ or $l = 1$: If there exists a set, say without loss of generality $B_1$, such that $B_1 = B$ then $\text{cost}(B_1) = \text{cost}(B)$. Since no vertex in $B$ is extendable (Proposition 10.6 item (8)) it holds that $\text{cost}(B_i) \geq 1$ for all $i \in [l]$. Now, if $l > 1$ and $B_1 = B$ then $\sum_{i=1}^{l} \text{cost}(B_i) \geq \text{cost}(B) + (l-1) > \text{cost}(B)$, which contradicts the assumption. Thus, we have either $l = 1$ and $B_1 = B$ (since $B_1$ covers $B$) or that the sets $B_1, B_2, \ldots, B_l \subsetneq B$ cover $B$. Since $B$ is strongly beneficial, and the sets $B_1, B_2, \ldots, B_l$ cover $B$, it must hold that $\sum_{i=1}^{l} \text{cost}(B_i) > \text{cost}(B)$ or that $l = 1$. Therefore, $\sum_{i=1}^{l} \text{cost}(B_i) > \text{cost}(B)$ if $l > 1$ and $\sum_{i=1}^{l} \text{cost}(B_i) = \text{cost}(B)$ if $l = 1$, but, this must hold for all $1 \leq p < p \leq k$. Hence, we must always have the latter case where $l = 1$, and there exist no $1 \leq \bar{p} < \bar{q} \leq k$ such that $\sum_{h \in [s]^d} \sum_{e \in E(G_h) \cap E(V_{\bar{p}}, V_{\bar{q}})} \text{cost}(B_h^e(F)) < \text{cost}(B)$. Thus, it holds for all $1 \leq p < q \leq k$ that there exists an $\hat{h} \in [s]^d$ and an edge $\hat{e} \in E(G_{\hat{h}}) \cap E(V_p, V_q)$ such that $B_{\hat{h}}^{\hat{e}}(F) = B_{\hat{h}}^{\hat{e}}$; such a set exists, because $B''$ contains at least one copy of each vertex in $B$ and $l = 1$. Furthermore, all other sets $B_h^e(F)$, with $h \in [s]^d$, $e \in E(G_h) \cap E(V_p, V_q)$, and $h \neq \hat{h}$ or $e \neq \hat{e}$, are empty since $l = 1$. We will show that $\hat{h} = h^*$ and that $\hat{e} = \{x_p, x_q\}$:

Consider vertex $x^{p,q}_{i,h^*_i}$, with $i \in [d-1]$, which is contained in the set $X_{i,h^*_i}$. This vertex must be covered by an edge $f$ in $F$ and the other endpoint of this edge $f$ must be contained in the set $\{b^e_{h,i} \mid h \in [s]^d, h_i = h^*_i, e \in E(G_h) \cap E(V_p, V_q)\}$ (see proof of Claim 10.25). This vertex must also be contained in the set $B^{\hat{e}}_{\hat{h}}$ because all other sets $B^e_h(F)$ with $e \in E(V_p, V_q)$ are empty; thus $\hat{h}_i = h^*_i$. This holds for all $i \in [d-1]$ which implies that $(\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_{d-1}) = (h^*_1, h^*_2, \ldots, h^*_{d-1})$. Furthermore, vertex $y^{\{x_p,x_q\}}_{h^*_d}$ must be covered by an edge $f \in F$ whose other endpoint is contained in the set $\{b^{\{x_p,x_q\}}_{h,d} \mid h \in [s]^d, h_d = h^*_d, \{x_p, x_q\} \in E(G_h) \cap E(V_p, V_q)\}$. For the same reasons, this vertex must be contained in the set $B^{\hat{e}}_{\hat{h}}$. The only vertex in the set $\{b^{\{x_p,x_q\}}_{h,d} \mid h \in [s]^d, h_d = h^*_d, \{x_p, x_q\} \in E(G_h) \cap E(V_p, V_q)\}$ that is also contained in $B^{\hat{e}}_{\hat{h}}$ is vertex $b^{\{x_p,x_q\}}_{h^*,d}$. Thus, $\hat{h}_d = h^*_d$, $\hat{e} = \{x_p, x_q\}$, and $\{y^{\{x_p,x_q\}}_{h^*_d}, b^{\{x_p,x_q\}}_{h^*,d}\}$ is an edge in $G'$. Summarized, we showed that $B^e_h(F) = B^e_h$ if and only if $h = h^*$ and $e \in E(X, X)$, and that all other sets $B^e_h(F)$ are empty.

We will show that the vertex set $X = \{x_1, x_2, \ldots, x_k\}$ is a clique in $G_{h^*}$. Recall that the vertex $x_i$ is contained in $V_i$, thus every vertex of $X$ is contained in a different color class. Consider two vertices $x_p$, $x_q$ with $1 \le p < q \le k$. We have to show that $\{x_p, x_q\}$ is an edge in $E(G_{h^*})$. Since vertex $y^{\{x_p,x_q\}}_{h^*_d}$ is adjacent to vertex $b^{\{x_p,x_q\}}_{h^*,d}$ it holds that $\{x_p, x_q\}$ is an edge in $G_{h^*}$, which proves that $X$ is a clique in $G_{h^*}$. ■

Concluding the section, we observe a simple quadratic lower bound for the size of kernels for Edge Dominating Set parameterized by the size of a modulator to $\mathcal{H}$-component graphs that holds for all sets $\mathcal{H}$.

**Lemma 10.27.** *The* Edge Dominating Set *problem has no kernelization to size $\mathcal{O}(n^{2-\varepsilon})$ where $n$ is the number of vertices, for any $\varepsilon > 0$, unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. *Therefore, for any set $\mathcal{H}$ of (connected) graphs, the* Edge Dominating Set *problem parameterized by the size of a modulator to $\mathcal{H}$-component graphs admits no kernelization to size $\mathcal{O}(|X|^{2-\varepsilon})$, for any $\varepsilon > 0$, unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

**Proof.** It is known that Vertex Cover admits no kernelization to size $\mathcal{O}(n^{2-\varepsilon})$, for any $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [DvM14]. By a straightforward reduction to an EDS instance $(G', k)$ with $n' = \mathcal{O}(n)$ vertices the same is true for EDS. This in turn yields an equivalent instance $(G', k, X')$ with a trivial modulator $X' = V(G')$ such that $G' - X'$ is the empty graph; since this is a feasible instance for the Edge Dominating Set problem parameterized by the size of a modulator to $\mathcal{H}$-component graphs for all sets $\mathcal{H}$ and since $|X'| = n' = \mathcal{O}(n)$, the lemma follows.

The lower bound of $\mathcal{O}(n^{2-\varepsilon})$ for Edge Dominating Set is not surprising (and may be well-known), as the same is known for a number of similar graph problems (like Vertex Cover). Thus, we only sketch a simple reduction (which surely has been rediscovered several times already).

Let $(G, k)$ be an instance of Vertex Cover with $G = (V, E)$ and, without loss of generality, $k \le |V|$. Construct a graph $G'$, starting from a copy of $G$ by adding

$2k$ vertices $u_1, \dots, u_k, u'_1, \dots, u'_k$ and adding the edges $\{u_1, u'_1\}, \dots, \{u_k, u'_k\}$. Finally, make each vertex $u_i$ adjacent to all vertices in the copy of $V$ in $G'$. Return the instance $(G', k)$. Clearly, $G'$ has $n + 2k = \mathcal{O}(n)$ vertices and the construction can be done in polynomial time.

It is easy to see that $(G, k)$ is a yes-instance for Vertex Cover if and only if $(G', k)$ is a yes-instance for Edge Dominating Set: If $(G, k)$ is a yes-instance for Vertex Cover then we can pick a vertex cover $S = \{v_1, \dots, v_k\}$ of size exactly $k$. Clearly, $F = \{\{v_1, u_1\}, \dots, \{v_k, u_k\}\}$ is an edge dominating set of size $k$ for $G'$, because all additional edges in comparison to $G$ have an endpoint in $\{u_1, \dots, u_k\}$. For the converse, assume that $(G', k)$ is a yes-instance and let $F$ be an edge dominating set of size $k$. Let $S$ contain all vertices of $V$ in $G'$ that are endpoints of $F$. Observe that, because $F$ needs to contain at least one vertex per edge $\{u_1, u'_1\}, \dots, \{u_k, u'_k\}$, which are disjoint from $V$, and because it has at most $2k$ endpoints, the set $S$ has size at most $k$. Clearly, the set $S$ alone covers all edges of $G'[V] \cong G$, so $(G, k)$ is a yes-instance for Vertex Cover. This completes the proof. ∎

## 10.5. Proof of Proposition 10.6

Let $H = (V, E)$ be a connected graph, let $W = W(H)$ be the set of free vertices, let $Q = Q(H)$ be the set of extendable vertices, and let $U = U(H)$ be the set of uncovered vertices.

**Lemma 10.28 (Proposition 10.6 (1)).** *The set $W$ is well defined.*

**Proof.** To show that the maximum free set is unique, we show that the union of two free sets is free. This implies that $W$ is the union of all free sets in $H$. Let $Y_1, Y_2 \subseteq Q$ be free sets in $H$, and let $Y = Y_1 \cup Y_2$. Since $Y_1$ and $Y_2$ are free, it hold that for all $y \in Y$ and for all minimum edge dominating set $F$ in $H$ there exists a minimum edge dominating set $F'$ in $H - y$ (of size $|F| - 1$) with either $V(F) \subseteq V(F') \setminus Y_1 \subseteq V(F') \setminus Y$ (if $y \in Y_1$) or $V(F) \subseteq V(F') \setminus Y_2 \subseteq V(F') \setminus Y$ (if $y \in Y_2$); thus $Y$ is free. ∎

**Lemma 10.29 (Proposition 10.6 (2)).** *The set $U$ is an independent set and no vertex in $Q$ is adjacent to a vertex in $U$, i.e., $N_H(U) \cap (Q \cap U) = \emptyset$.*

**Proof.** If there was an edge $\{u, u'\}$ with $u, u' \in U$ then no feasible edge dominating set could avoid being incident with either vertex. If $u \in U$ had a neighbor $q \in Q$ then $\textsc{eds}(H - q) + 1 = \textsc{eds}(H)$ but then combining a minimum solution for $H - q$ and adding edge $\{u, q\}$ would be a minimum solution for $H$ and be incident with $u$; a contradiction. ∎

**Lemma 10.30 (Proposition 10.6 (3)).** *If $v \in N_H(U)$ is a vertex that is adjacent to a vertex in $U$, then $v$ is contained in every minimum edge dominating set of $H$*

**Proof.** Vertices in $U$ are never endpoints of edges in minimum solutions. Thus, to dominate the incident edges all their neighbors must be endpoints of solution edges. ■

**Lemma 10.31 (Proposition 10.6 (4)).** *It holds for all vertices $v \in V$ that $\textsc{eds}(H)-1 \leq \textsc{eds}(H-v) \leq \textsc{eds}(H)$.*

**Proof.** Let $F_v$ be an edge dominating set of $H-v$ and let $u \in V$ such that $\{u,v\} \in E$. Clearly, $F_v \cup \{\{u,v\}\}$ is an edge dominating set of $H$; hence $\textsc{eds}(H) - 1 \leq |F_v| = \textsc{eds}(H-v)$. Now, let $F$ be a minimum edge dominating set of $H$. If $v$ is not incident with an edge in $F$, then $F$ is also an edge dominating set in $H$; hence $\textsc{eds}(H-v) \leq |F| = \textsc{eds}(H)$. If $v$ is incident with an edge $f = \{v,u\}$ in $F$, then replace $f$ either by an edge in $\delta_H(v)\setminus\{f\}$ or delete $f$ when $\delta_H(v)\setminus\{f\}$ is empty; hence $\textsc{eds}(H-v) \leq |F| = \textsc{eds}(H)$. ■

**Lemma 10.32 (Proposition 10.6 (5)).** *Let $Y \subseteq V$. It holds for all subsets $X \subseteq Y$ that $\textsc{eds}(H-X) - |Y \setminus X| \leq \textsc{eds}(H-Y) \leq \textsc{eds}(H-X)$, and that $\operatorname{cost}(X) \leq \operatorname{cost}(Y)$.*

**Proof.** It follows from Proposition 10.6 (4) that $\textsc{eds}(H-X)-|Y\setminus X| \leq \textsc{eds}(H-Y) \leq \textsc{eds}(H-X)$: Let $Y = \{y_1, y_2, \ldots, y_p\}$ and let $X = \{y_1, y_2, \ldots, y_q\}$ with $q \leq p$. It holds that:

$$\textsc{eds}(H-Y) \leq \textsc{eds}(H-(Y\setminus\{y_p\})) \leq \textsc{eds}(H-(Y\setminus\{y_{p-1},y_p\})) \leq \ldots \leq \textsc{eds}(H-X).$$

Furthermore, we can obtain that

$$\begin{aligned}\textsc{eds}(H-Y) \geq \textsc{eds}(H-(Y\setminus\{y_p\})) - 1 \geq \textsc{eds}(H-(Y\setminus\{y_{p-1},y_p\})) - 2 \geq \\ \ldots \geq \textsc{eds}(H-X) - |Y\setminus X|.\end{aligned}$$

Now, combining the above

$$\begin{aligned}\operatorname{cost}(Y) &= \textsc{eds}(H-Y) + |Y| - \textsc{eds}(H) \geq \textsc{eds}(H-X) - |Y\setminus X| + |Y| - \textsc{eds}(H) \\ &= \textsc{eds}(H-X) + |X| - \textsc{eds}(H) = \operatorname{cost}(X).\end{aligned}$$ ■

**Lemma 10.33 (Proposition 10.6 (6)).** *Let $F$ be a minimum edge dominating set in $H$. There exists a minimum edge dominating set $F'$ in $H$ with $(V(F)\cup N_H(W))\setminus W \subseteq V(F')$.*

**Proof.** Let $F$ be an arbitrary minimum edge dominating set in $H$, and let $F'$ be a minimum edge dominating set in $H$ with $V(F) \setminus W \subseteq V(F')$ and $|V(F') \cap N_H(W)|$ maximal (under the minimum edge dominating sets that fulfill $V(F) \setminus W \subseteq V(F')$). Assume for contradiction that $V(F') \cap N_H(W) \neq N_H(W)$. Let $v \in N_H(W) \setminus V(F')$ be a vertex in the neighborhood of $W$ that is not incident with an edge in $F'$, and let $w \in N_H(v) \cap W$ be a free vertex that is adjacent to $v$. Since vertex $w$ is free, there exists a minimum edge dominating set $\widetilde{F}$ in $H-w$ (of size $|F'|-1$) with $V(F') \setminus W \subseteq V(\widetilde{F})$.

Now, we can add the edge $\{v, w\}$ to the minimum edge dominating set $\widetilde{F}$ to obtain a minimum edge dominating set $\hat{F} = \widetilde{F} \cup \{\{v, w\}\}$ of $H$. It holds that $V(F) \setminus W \subseteq V(F') \setminus W \subseteq V(\widetilde{F}) \setminus W \subseteq V(\hat{F}) \setminus W$. Furthermore, the set $V(F') \cap N_H(W)$ is a proper subset of $V(\hat{F}) \cap N_H(W)$, because $V(\hat{F})$ contains all vertices in $V(F') \setminus W$ and the vertex $v \notin W$ that is not contained in $V(F')$. This contradicts the choice of $F'$ and proves the statement. ■

**Lemma 10.34 (Proposition 10.6 (7)).** *Every set that consists of a single vertex $v \in Q \setminus W$ is strongly beneficial. Furthermore, these are the only beneficial sets of size one.*

**Proof.** Let $v \in Q \setminus W$ be a vertex that is extendable and not free, and let $B = \{v\}$. We show that $B$ is strongly beneficial. Since $\textsc{eds}(H - v) + 1 = \textsc{eds}(H)$, and since for every set $\widetilde{B} \subsetneq B$ it holds that $\textsc{eds}(H - \widetilde{B}) = \textsc{eds}(H)$ it holds that $B$ is beneficial. ($\widetilde{B} = \emptyset$ is the only proper subset of $B$.) Assume for contradiction that $B$ is not strongly beneficial. This would imply that there exists a cover $B_1, B_2, \ldots, B_h \subsetneq B$, but the only proper subset of $B = \{v\}$ is the empty set. Thus, $B$ is strongly beneficial.

Assume there exists a beneficial set $B = \{v\}$ with $v \notin Q \setminus W$. Note that beneficial sets are disjoint from $W$ (by definition), thus $v \in V \setminus Q$. Since $B$ is beneficial it holds that $\textsc{eds}(H - B) < \textsc{eds}(H)$. Together with Proposition 10.6 item (4) it follows that $\textsc{eds}(H - B) = \textsc{eds}(H) - 1$; thus $v \in Q$ which is a contradiction. ■

**Lemma 10.35 (Proposition 10.6 (8)).** *If $B$ is a strongly beneficial set of size at least two then $B$ contains no extendable vertex, i.e., $B \cap Q = \emptyset$.*

**Proof.** If there would exist a vertex $v \in B$ that is extendable, but not free, then $\mathrm{cost}(\{v\}) = \textsc{eds}(H - v) + |\{v\}| - \textsc{eds}(H) = 0$. Furthermore, it follows from Proposition 10.6 item (5) that $\mathrm{cost}(B \setminus \{v\}) \leq \mathrm{cost}(B)$. Now, $\{v\}, B \setminus \{v\} \subsetneq B$ is a cover of $B$ and it holds that $\mathrm{cost}(\{v\}) + \mathrm{cost}(B \setminus \{v\}) \leq 0 + \mathrm{cost}(B) = \mathrm{cost}(B)$. This implies that $B$ is not strongly beneficial which is a contradiction. ■

**Lemma 10.36 (Proposition 10.6 (9)).** *If there exists a set $Y \subseteq V \setminus W$ such that $\textsc{eds}(H - Y) < \textsc{eds}(H)$, then there exists a beneficial set $B \subseteq Y$ with $\textsc{eds}(H - B) = \textsc{eds}(H - Y)$.*

**Proof.** If $Y$ is a beneficial set then $B = Y$ is a beneficial set with $\textsc{eds}(H - Y) = \textsc{eds}(H - B)$. Thus, assume that $Y$ is not beneficial. Hence, there exists a set $Y' \subsetneq Y$ with $\textsc{eds}(H - Y) \geq \textsc{eds}(H - Y')$ (by definition of beneficial). Pick $B \subsetneq Y$ minimal with $\textsc{eds}(H - B) \leq \textsc{eds}(H - Y)$. This implies that $B$ is beneficial: Otherwise there would exist a set $Y' \subsetneq B$ with $\textsc{eds}(H - Y') \leq \textsc{eds}(H - B)$ which contradicts the choice of $B$. Since $B \subseteq Y$ it follows from Proposition 10.6 item (5) that $\textsc{eds}(H - B) = \textsc{eds}(H - Y)$. ■

**Lemma 10.37 (Proposition 10.6 (10)).** *If there exists a set $Y \subseteq V \setminus W$ such that $\textsc{eds}(H - Y) < \textsc{eds}(H)$, then there exists a beneficial set $B \subseteq Y$ with $\textsc{eds}(H - B) + 1 = \textsc{eds}(H)$. Furthermore, $B$ is strongly beneficial.*

**Proof.** Let $B \subseteq Y$ be minimal such that $\text{EDS}(H - B) < \text{EDS}(H)$. Thus, for every $Z \subsetneq B$ it holds $\text{EDS}(H - Z) = \text{EDS}(H)$. First, we prove that $B$ is beneficial. If $B$ is not beneficial, then there exists a set $\widetilde{B} \subsetneq B$ such that $\text{EDS}(H - B) \geq \text{EDS}(H - \widetilde{B})$. This contradicts the choice of $B$, because $\widetilde{B} \subsetneq B$ and $\text{EDS}(H - \widetilde{B}) < \text{EDS}(H)$; hence $B$ is beneficial.

Next, we show that $\text{EDS}(H - B) + 1 = \text{EDS}(H)$. Let $b \in B$ and $B' = B \setminus \{b\}$. It holds that $\text{EDS}(H - B') = \text{EDS}(H)$ (choice of $B$). It follows from Proposition 10.6 item (4) that $\text{EDS}(H) - 1 = \text{EDS}(H - B') - 1 \leq \text{EDS}(H - B' - b) = \text{EDS}(H - B) \leq \text{EDS}(H - B') = \text{EDS}(H)$. Since $\text{EDS}(H - B) < \text{EDS}(H)$ it follows that $\text{EDS}(H - B) + 1 = \text{EDS}(H)$.

Finally, we show that $B$ is strongly beneficial. If $B$ has size one, then $B = \{v\}$ with $v \in Q \setminus W$, and it follows from Proposition 10.6 item (7) that $B$ is strongly beneficial. Now, assume for contradiction that $B$ is not strongly beneficial. Hence, there exists a cover $B_1, B_2, \ldots, B_h \subsetneq B$ of $B$ with $\text{cost}(B) \geq \sum_{i=1}^{h} \text{cost}(B_i)$. It holds that $\text{cost}(B) = \text{EDS}(H-B) + |B| - \text{EDS}(H) = |B| - 1$ (because $\text{EDS}(H-B)+1 = \text{EDS}(H)$), and it always holds that $\text{cost}(B_i) \leq |B_i|$ (definition of cost). This implies that there exists at least one $i^* \in [h]$ with $\text{cost}(B_{i^*}) < |B_{i^*}|$: otherwise

$$|B| - 1 = \text{cost}(B) \geq \sum_{i=1}^{h} \text{cost}(B_i) = \sum_{i=1}^{h} |B_i| \geq |B|.$$

Now, $|B| - \text{cost}(B) = 1 \leq |B_{i^*}| - \text{cost}(B_{i^*})$. This is a contradiction to the fact that $B$ is beneficial, because $B_{i^*} \subsetneq B$ is a proper subset of $B$ and it holds that $|B| - \text{cost}(B) \leq |B_{i^*}| - \text{cost}(B_{i^*})$. Thus, $B$ is strongly beneficial. ∎

**Lemma 10.38 (Proposition 10.6 (11)).** *If $H$ has a beneficial set $B$, then $H$ has also a strongly beneficial set $B' \subseteq B$.*

**Proof.** This follows from Proposition 10.6 item (10). ∎

**Lemma 10.39 (Proposition 10.6 (12)).** *Let $F$ be a minimum edge dominating set in $H$. If $e = \{x, y\}$ is an edge in $F$ with $x, y \notin Q$, then $\{x, y\}$ is a strongly beneficial set in $H$.*

**Proof.** Let $F$ be a minimum edge dominating set in $H$ and let $\{x, y\}$ be an edge in $F$ with $x, y \notin Q$. First, we show that $B = \{x, y\}$ is beneficial. It holds that $\text{EDS}(H - B) < \text{EDS}(H)$, because $F' = F \setminus \{\{x, y\}\}$ is an edge dominating set in $H - B$: $F'$ covers all edges that are not incident with $x$ and $y$, and these vertices are not contained in $H - B$. It follows from Proposition 10.6 item (10) that there exists a strongly beneficial set $B' \subseteq B$ with $\text{EDS}(H - B') + 1 = \text{EDS}(H)$. The sets $\emptyset$, $\{x\}$ and $\{y\}$ are not beneficial, because the empty set is not beneficial, and neither $x$ nor $y$ is extendable; hence $\text{EDS}(H - x) = \text{EDS}(H - y) = \text{EDS}(H)$. Note that the only beneficial sets of size one consist of one vertex in $Q \setminus W$ (Proposition 10.6 item (7)). Thus, $B$ must be strongly beneficial. ∎

**Lemma 10.40 (Proposition 10.6 (13)).** *Let $B$ be a beneficial set. $B$ is strongly beneficial if and only if for every non-trivial partition $B_1, B_2, \dots, B_h$ of $B$ it holds that $\operatorname{cost}(B) < \sum_{i=1}^{h} \operatorname{cost}(B_i)$.*

**Proof.** ($\Rightarrow$:) This follows directly from the definition of strongly beneficial sets, because every non-trivial partition $B_1, B_2, \dots, B_h$ of $B$ is also a cover of $B$ with $B_1, \dots, B_h \subsetneq B$.

($\Leftarrow$:) Assume that $B$ is not strongly beneficial. This implies that there exists a cover $B_1, B_2, \dots, B_h \subsetneq B$ of $B$ with $\operatorname{cost}(B) \geq \sum_{i=1}^{h} \operatorname{cost}(B_i)$. We construct a non-trivial partition of $B$ as follows: Let $B'_1 = B_1$ and let $B'_i = B_i \setminus \left(\bigcup_{j=1}^{i-1} B_j\right)$ for all $2 \leq i \leq h$. It holds that no set $B'_i$ is the set $B$ because $B'_i \subseteq B_i \subsetneq B$. Furthermore, the union of all sets $B'_i$, with $i \in [h]$, is still $B$, and the intersection of two sets $B_i$ and $B_j$ with $i \neq j$ is empty (by construction). Thus, all non-empty sets $B'_i$ are a non-trivial partition of $B$. Additionally, it follows from Proposition 10.6 item (5) that $\operatorname{cost}(B'_i) \leq \operatorname{cost}(B_i)$ which implies that $\operatorname{cost}(B) \geq \sum_{i=1}^{h} \operatorname{cost}(B_i) \geq \sum_{i=1}^{h} \operatorname{cost}(B'_i)$. Hence, there exists a non-trivial partition $B''_1, B''_2, \dots, B''_q$ of $B$ with $\operatorname{cost}(B) \geq \sum_{i=1}^{q} \operatorname{cost}(B''_i)$. This concludes the proof. ∎

**Lemma 10.41 (Proposition 10.6 (14)).** *Let $Y \subseteq V \setminus W$. There exists a partition $B_1, B_2, \dots, B_h$ of $Y$ where $B_i$ is either strongly beneficial or where $B_i$ has $\operatorname{cost}(B_i) = |B_i|$, for all $i \in [h]$, such that $\operatorname{cost}(Y) \geq \sum_{i=1}^{h} \operatorname{cost}(B_i)$. (Note that we also allow trivial partitions.)*

**Proof.** Assume that the statement does not hold and let $Y \subseteq V \setminus W$ be a minimal set that does not fulfill the properties of the lemma. Hence, $Y$ is neither strongly beneficial nor has $\operatorname{cost}(Y) = |Y|$, because in both cases the trivial partition $Y$ would fulfill the properties of the lemma.

First, assume that $Y$ is not beneficial. It follows from Proposition 10.6 item (9) that there exists a set $Y' \subsetneq Y$ beneficial with $\textsc{eds}(H - Y) = \textsc{eds}(H - Y')$. Since $Y' \subsetneq Y$ is a proper subset of $Y$ and $Y$ is a minimal set that does not fulfill the properties of the lemma, there exists a partition $B'_1, B'_2, \dots, B'_p$ of $Y'$ where $B'_i$ is either strongly beneficial or has $\operatorname{cost}(B'_i) = |B_i|$ for all $i \in [p]$ such that $\operatorname{cost}(Y') \geq \sum_{i=1}^{p} \operatorname{cost}(B'_i)$. Furthermore, the set $Y'' = Y \setminus Y'$ is a proper subset of $Y$, because $Y'$ is not the empty set (if $Y'$ is the empty set then $Y'$ is not beneficial). Thus, there exists a partition $B''_1, B''_2, \dots, B''_q$ of $Y''$ where $B''_i$ is either strongly beneficial or has $\operatorname{cost}(B''_i) = |B''_i|$ for all $i \in [q]$ such that $\operatorname{cost}(Y'') \geq \sum_{i=1}^{q} \operatorname{cost}(B''_i)$.

Now, $B_1 = B'_1, B_2 = B'_2, \dots, B_p = B'_p, B_{p+1} = B''_1, B_{p+2} = B''_2, \dots, B_{p+q} = B''_q$ is a partition of $Y$, because $B'_1, B'_2 \dots, B'_p$ is a partition of $Y'$, $B''_1, B''_2, \dots, B''_q$ is a partition of $Y''$, and $Y', Y''$ is a partition of $Y$. Additionally, every set $B_i$, with $i \in [p+q]$, is either strongly beneficial or has $\operatorname{cost}(B_i) = |B_i|$ (by choice of $B_i$). To show that $Y$ also fulfills the properties of the lemma it remains to show that $\operatorname{cost}(Y) \geq \sum_{i=1}^{p+q} \operatorname{cost}(B_i)$.

It holds that:

$$\begin{aligned}
\operatorname{cost}(Y) &= \textsc{eds}(H - Y) + |Y| - \textsc{eds}(H) \\
&= \textsc{eds}(H - Y') + |Y'| + |Y''| - \textsc{eds}(H) && \text{, by choice of } Y' \text{ and } Y = Y'\dot{\cup}Y'' \\
&\geq \operatorname{cost}(Y') + \operatorname{cost}(Y'') && \text{, by the definition of cost} \\
&\geq \sum_{i=1}^{p} \operatorname{cost}(B_i') + \sum_{i=1}^{q} \operatorname{cost}(B_i'') = \sum_{i=1}^{p+q} \operatorname{cost}(B_i).
\end{aligned}$$

This implies that $Y$ fulfills the properties of the lemma, which is a contradiction.

Thus, assume that $Y$ is beneficial (but not strongly beneficial). Hence, there exists a non-trivial partition $B_1, B_2, \ldots, B_h$ of $Y$ with $\operatorname{cost}(Y) \geq \sum_{i=1}^{h} \operatorname{cost}(B_i)$ Proposition 10.6 item (13). Every $B_i$, with $i \in [h]$, is a proper subset of $Y$, because $B_1, B_2, \ldots, B_h$ is a non-trivial partition of $Y$. Since $Y$ is a minimal set that does not fulfill the properties of the lemma, there exists, for all $i \in [h]$, a partition $B_{i,1}, B_{i,2}, \ldots, B_{i,p_i}$ of $B_i$ where $B_{i,j}$ is either strongly beneficial or has $\operatorname{cost}(B_{i,j}) = |B_{i,j}|$, for all $j \in [p_i]$, such that $\operatorname{cost}(B_i) \geq \sum_{j=1}^{p_i} \operatorname{cost}(B_{i,j})$. By construction, the sets $B_{1,1}, B_{1,2}, \ldots, B_{1,p_1}, B_{2,1}, \ldots, B_{h,p_h}$ are a partition of $Y$ where every $B_{i,j}$ is either strongly beneficial or has $\operatorname{cost}(B_{i,j}) = |B_{i,j}|$, for all $i \in [h]$ and all $j \in [p_i]$. Furthermore, $\operatorname{cost}(Y) \geq \sum_{i=1}^{h} \operatorname{cost}(B_i) \geq \sum_{i=1}^{h} \sum_{j=1}^{p_i} \operatorname{cost}(B_{i,j})$. Thus, the set $Y$ fulfills the properties of the lemma, which is a contradiction and concludes the proof. ■

## 10.6. Summary

We showed that our lower bound construction for Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph can be extended to a lower bound for Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph if $\mathcal{H}$ contains a graph $H$ that has a control pair. Since the path $P = v_1 e_1 v_2 e_2 v_3$ has the control pair $(\{v_1\}, \{v_2\})$ and the triangle $T = w_1 e_1 w_2 e_2 w_3 e_3 w_1$ has the control pair $(\{w_1\}, \{w_2, w_3\})$ this result implies that for all hereditary graph classes $\mathcal{C}$, except $\mathcal{C}$ being a class where each connected component has one or two vertices, Edge Dominating Set parameterized by the size of a $\mathcal{C}$-modulator has no polynomial kernel, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Recall, that we showed in Chapter 9 that Edge Dominating Set parameterized by the size of a degree-1-modulator has a polynomial kernel.

However, we were able to also extend the polynomial kernel for Edge Dominating Set parameterized by the size of a modulator to a $P_5$-component graph to Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph if $\mathcal{H}$ is finite and no graph in $\mathcal{H}$ contains a control pair. Overall, we gave a complete classification for Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph for all finite sets $\mathcal{H}$.

# CHAPTER 11
# CONCLUSION AND OPEN PROBLEMS

In this chapter we summarize the results of this part and we discuss some open problems. Motivated by the positive results of Part II we tried to figure out for which structural parameters Edge Dominating Set admits a polynomial kernel. We observed that the structures that causes hardness are much more complicated than for Vertex Cover where for every graph class $\mathcal{C}$ that has only constant size components it holds that Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator has a polynomial kernel. First of all, we showed that Edge Dominating Set parameterized by the size of a modulator to a $P_3$-component graph has no polynomial kernel (unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$) and that Edge Dominating Set parameterized by the size of a degree-1-modulator has a polynomial kernel. Furthermore, we showed that Edge Dominating Set parameterized by the above lower bound $\frac{1}{2}$mm is para-$\mathsf{NP}$-complete.

Finally, we gave a complete classification for Edge Dominating Set parameterized by the size of a modulator to $\mathcal{H}$-component graphs for all finite sets $\mathcal{H}$. More precisely, we showed that if there exists a graph in $\mathcal{H}$ that has a control pair then Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Moreover, we showed that all graphs that have no control pair have a very specific structure and that Edge Dominating Set parameterized by the size of a modulator to an $\mathcal{H}$-component graph, where no graph in $\mathcal{H}$ has a control pair, has a polynomial kernel. The size depends on the largest strongly beneficial set.

Our classification shows that Edge Dominating Set parameterized by the size of a modulator to a $P_3$- as well as to a $K_3$-component graph (and thus also by the size of a $\mathcal{C}$-modulator, with $P_3 \in \mathcal{C}$ or $K_3 \in \mathcal{C}$) has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Consequently, the unique maximal hereditary graph class $\mathcal{C}$ for which a parameterization by the size of a $\mathcal{C}$-modulator possibly has a polynomial kernel, is the class of graphs where every vertex has degree at most one. As shown in Section 9.3, Edge Dominating Set parameterized by the size of a degree-1-modulator has a polynomial kernel.

**Open problems.** An obvious follow-up question is to extend our result to infinite sets $\mathcal{H}$. Our lower bounds of course continue to work in this setting, and the upper bounds still permit us to reduce the number of connected components (under the same conditions as before, e.g., that relevant beneficial sets have bounded size). However, for infinite $\mathcal{H}$, polynomial kernels also require us to shrink connected components of $G - X$, and to derive general rules for this. Moreover, even determining beneficial sets for graphs $H \in \mathcal{H}$ could no longer be dismissed as being constant time. It is conceivable that such a classification is doable whenever the graphs in $\mathcal{H}$ have bounded treewidth, as this simplifies the required additional steps. This seems to be a reasonable goal, because most known tractable graph classes for Edge Dominating Set have bounded treewidth and tractability for $G - X$ is required, or else NP-hardness for $|X| = 0$ rules out kernels and fixed-parameter tractability.

A suitable set $\mathcal{H}$ to start with could be the set of graphs containing all paths of length $3\mathbb{N}_0 + 1$, i.e., with $3\mathbb{N}_0 + 2$ vertices. When $\mathcal{H}$ contains only paths of length $1, 4, 7, \dots, 3c + 1$ for a constant $c$, we know that there exists a kernel with $\mathcal{O}(|X|)$ vertices. Besides, we know exactly which vertices of a path of length $3\mathbb{N}_0 + 1$ are free and that these paths have neither uncovered vertices nor beneficial sets. Is it possible to shrink the connected components of $G - X$ that are isomorphic to a path of length $3\mathbb{N}_0 + 1$?[1] Apart from this, it would be nice to close the gap between the kernel size $\mathcal{O}(|X|^{d+1} \log |X|)$ and the lower bound of $\mathcal{O}(|X|^{d-\varepsilon})$, where improvements to the upper bound seem more likely.

Regarding above lower bounds, the NP-hardness for Edge Dominating Set on graphs with a perfect matching of size MM, even for $k = \frac{1}{2}\text{MM}$, leaves little hope for tractability above tight lower bounds. Still, are there other nontrivial, perhaps non-tight, lower bounds $L(G)$ such that we get at least fixed-parameter tractability for parameter $\ell = k - L(G)$? Another interesting parameter could be the “below upper bound” parameter $\ell = \text{MM} - k$. At least, it holds that Edge Dominating Set is polynomial-time solvable for $\ell = 0$.

However, the probably most important question, when it comes to polynomial kernels for Edge Dominating Set, is, whether Edge Dominating Set parameterized by solution size has a kernel of size $\mathcal{O}(k^2)$. We know that we can reduce to $\mathcal{O}(k^2)$ vertices and $\mathcal{O}(k^3)$ edges, but we only know that there exists no kernel of size $\mathcal{O}(k^{2-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

[1] Another possible class would be the class of cliques of even size.

# Part IV.

# Subset Feedback Vertex Set

**– The Mysterious Problem –**

# CHAPTER 12 INTRODUCTION TO GRAPH CUT PROBLEMS

## 12.1. Graph Cut and Feedback Problems

In a *graph cut* problem, we are given a directed or undirected graph, an integer $k$, and we need to determine whether there exists a *cutset*[1] of size at most $k$ whose removal makes the graph satisfy some global separation requirements. The most basic graph cut problem is the classical minimum $(s,t)$-cut problem, where given a directed graph $G$, two distinct vertices $s,t \in V(G)$, an integer $k$, and we want to determine whether there exists a set $F$ of at most $k$ edges such $G-F$ does not contain any $s,t$-path. Ford and Fulkerson [FF56, FF62] showed that one can find such a set in polynomial time. It is well known that one can extend this result to undirected graphs as well as to deleting vertices instead of edges.

There are two natural and well-studied generalizations of the minimum $(s,t)$-vertex cut problem in undirected graphs where the cutset is a set of vertices, namely the Multiway Cut problem (MWC) and the Multicut problem (MC). In the Multiway Cut problem we are given an undirected graph $G$, a set $T = \{t_1, t_2, \ldots, t_s\}$ of *terminals*, an integer $k$, and the task is to determine whether there exists a cutset of at most $k$ non-terminal vertices whose deletion disconnects the terminals from one another. This problem is already NP-hard for three terminals [DJP+94]. In the Multicut problem we are also given an undirected graph $G$, an integer $k$, but instead of a set of terminals we are given a set of terminal pairs $\mathcal{T} = \{(s_i, t_i) \mid 1 \le i \le r\}$ and we need to determine whether there exists a cutset of at most $k$ non-terminal vertices such that after deleting the cutset from $G$, the terminal vertex $t_i$ is no longer reachable from the terminal vertex $s_i$ for all $1 \le i \le r$. Since Multiway Cut is NP-hard even when restricted to three terminals it follows that Multicut is already NP-hard for 3 terminal pairs, because there exists an easy reduction: Given an instance $(G,T,k)$ of Multiway Cut with

[1] A cutset is a set of vertices or edges, depending on the variant.

$T = \{t_1, t_2, t_3\}$ it holds that $(G, \mathcal{T}, k)$ with $\mathcal{T} = \{(t_1, t_2), (t_2, t_3), (t_3, t_1)\}$ is an instance of MULTICUT that is equivalent to the MULTIWAY CUT instance $(G, T, k)$. This is a standard and well-known polynomial (parameter) transformation from MULTIWAY CUT (parameterized by the solution size $k$) to MULTICUT (parameterized by the solution size $k$).

One important variant of MULTIWAY CUT is DELETABLE TERMINAL MULTIWAY CUT (DTMWC). Here, we are allowed to delete terminal vertices, more precisely, an instance of DELETABLE TERMINAL MULTIWAY CUT consists of an undirected graph $G$, an integer $k$, a set $T$ of terminals, and the question is whether there exists a cutset $X$ of at most $k$ vertices such that in $G - X$ no two terminals $t_1, t_2 \in T \setminus X$ are in the same connected component. This variant is still NP-hard, because one can reduce VERTEX COVER to this problem by setting $T = V(G)$. Furthermore, DELETABLE TERMINAL MULTIWAY CUT (parameterized by the solution size $k$) is easier than MULTIWAY CUT (parameterized by the solution size $k$), because there exists an easy polynomial (parameter) transformation from DELETABLE TERMINAL MULTIWAY CUT (parameterized by the solution size $k$) to MULTIWAY CUT (parameterized by the solution size $k$): Given an instance $(G, T, k)$ of DELETABLE TERMINAL MULTIWAY CUT, the instance $(G', T', k)$, where $G'$ results from $G$ by adding a copy $T'$ of $T$ to $G$ and connecting each $t \in T$ with its copy in $T'$, is an equivalent instance of MULTIWAY CUT.

As for the classical $(s, t)$-vertex cut problem, we can define MULTIWAY CUT and MULTICUT also on directed graphs: In the DIRECTED MULTIWAY CUT problem (DMWC) and the DIRECTED MULTICUT problem (DMC) we are given a directed graph $G$, a set $T$ of terminals or a set $\mathcal{T} = \{(s_i, t_i) \mid 1 \leq i \leq r\}$ of terminal pairs, respectively, an integer $k$, and have to decide whether there exists a cutset $X$ of at most $k$ non-terminal vertices such that in $G - X$ there is no directed path between any pair of terminals in $T$ or no directed path from $s_i$ to $t_i$ for $1 \leq i \leq r$, respectively. Garg et al. [GVY04] showed that DIRECTED MULTIWAY CUT is already NP-hard when restricted to two terminals. Since DIRECTED MULTIWAY CUT restricted to two terminals can be reduced to an instance of DIRECTED MULTICUT when restricted to two terminal pairs, it follows that DIRECTED MULTICUT is also NP-hard when restricted to two terminals pairs: Given an instance $(G, T = \{t_1, t_2\}, k)$ of DIRECTED MULTIWAY CUT restricted to two terminals, it holds that $(G, \mathcal{T} = \{(t_1, t_2), (t_2, t_1)\}, k)$ is an instance of DIRECTED MULTICUT that is equivalent to the DIRECTED MULTIWAY CUT instance $(G, T, k)$. Furthermore, when restricted to two terminals, respectively two terminal pairs, DIRECTED MULTIWAY CUT and DIRECTED MULTICUT are equivalent (cf. [CHM13]): It remains to show that we can reduce an instance $(G, \mathcal{T} = \{(s_1, t_2), (s_2, t_2)\})$ of DIRECTED MULTICUT to an instance of DIRECTED MULTIWAY CUT with two terminals. We construct a graph $G'$ by adding $k + 1$ copies of each terminal vertex, and denote the set of copies of $s_i$ and $t_i$ by $S_i$ and $T_i$ for $i = 1, 2$, respectively. Additionally, we add a vertex $s$ and a vertex $t$ to the graph and add the edges $(s, s'_1)$ with $s'_1 \in S_1$, $(t'_1, t)$ with $t'_1 \in T_1$, $(t, s'_2)$ with $s'_2 \in S_2$ and $(t'_2, s)$ with $t'_2 \in T_2$ to the graph. Denote the resulting graph by $G'$. Now, $(G, T = \{s, t\}, k)$ is an instance of DIRECTED MULTIWAY CUT that is equivalent to the DIRECTED MULTICUT instance $(G, \mathcal{T}, k)$. Observe, that $G'$ has an $s, t$-path (or

$t, s$-path) if and only if $G$ has an $s_1, t_1$-path (or $s_2, t_2$-path). This property still holds after removing vertices, since an optimum solution of size at most $k$ neither contains terminal vertices nor any vertex in $S_1 \cup S_2 \cup T_1 \cup T_2$.

Instead of choosing as cutset a set of vertices, we can also choose a set of edges. We call these problems, where we delete edges instead of vertices, Edge Multiway Cut and Edge Multicut. It is well known that choosing as cutset a set of vertices is more general than choosing as cutset a set of edges because we can easily reduce the edge version to the vertex version (cf. [Cun89]). However, for directed graphs these two versions are equivalent (cf. [CHM13, Section 2.1]).[2]

In a *feedback* problem we are given a directed or undirected graph $G$, a set $\mathcal{C}$ of cycles of $G$, an integer $k$, and we want to determine whether there exists a vertex set $X$, also called *transversal*, of size at most $k$ that intersects all cycles of $\mathcal{C}$.[3] The probably most studied feedback problem is Feedback Vertex Set (FVS), where given an undirected graph $G$, and an integer $k$, the objective is to decide whether there exists a vertex set $X$ of size at most $k$ that intersects all cycles of $G$, i.e., a set $X$ such that $G - X$ is a forest. Instead of asking whether there exists a vertex set of size at most $k$ that deletes all cycles of $G$, we could ask whether there exists a vertex set of size at most $k$ that intersects only all *odd* cycles, i.e., cycles of odd length. This well-known problem is called Odd Cycle Transversal (OCT).

A more general and more difficult version of the Feedback Vertex Set problem is the Subset Feedback Vertex Set problem, short Subset FVS or SFVS. Here, given an undirected graph $G$, a set $S$ of vertices, and an integer $k$, the task is to decide whether there exists a vertex set of size at most $k$ that hits all cycles through $S$. This problem is equivalent to the Edge Subset Feedback Vertex Set problem, short Edge Subset FVS, where $S$ is a set of edges instead of vertices [CPPW13b]. Note that an instance $(G, k)$ of Feedback Vertex Set, is equivalent to the instance $(G, S = V(G), k)$ of Subset FVS. Additionally, Cygan et al. [CPPW13b] showed that there exists a polynomial parameter transformation from Deletable Terminal Multiway Cut parameterized by the solution size to Edge Subset FVS parameterized by the solution size, i.e., Deletable Terminal Multiway Cut parameterized by the solution size is easier than Edge Subset FVS (and also Subset FVS) parameterized by the solution size. This polynomial parameter transformation constructs an instance $(G', S, k)$ of Edge Subset FVS from a given instance $(G, T, k)$ of Deletable Terminal Multiway Cut, by adding a copy $T'$ of $T$ to $G$ and by adding an edge between every vertex $t' \in T'$ and its corresponding vertex in $T$ as well as all other vertices in $T'$. The set $S$ consists of the set of edges between $T$ and $T'$.

Another well-studied feedback problem is Group Feedback Vertex Set, short Group FVS or GFVS, which generalizes Odd Cycle Transversal as well as Subset FVS, and is defined as follows: Let $(\Gamma, \cdot)$ be a group with identity element $1_\Gamma$, let

[2]In this paper they only show this for Directed Multiway Cut. However, the constructions also work for Directed Multicut.

[3]The set $\mathcal{C}$ of cycles must not be given explicitly, it only must be sufficiently defined.

$G = (V, E)$ be an undirected graph with orientation $\delta\colon E \to V \times V$ such that edge $e = \{x, y\} \in E$ is mapped either to $(x, y)$ or to $(y, x)$ via $\delta$; such a pair $(G, \delta)$ is called an *oriented graph*. A $\Gamma$*-labeling* of an oriented graph $(G = (V, E), \delta)$ consists of an assignment of a label $\gamma_e \in \Gamma$ for every edge $e \in E$, and a labeling function $\lambda\colon \{(e, v) \mid v \in e \in E\} \to \Gamma$ such that for each edge $e = \{u, v\} \in E$ with $\delta(e) = (u, v)$ it holds that $\lambda(e, u) = \gamma_e$ and $\lambda(e, v) = -\gamma_e$. For a cycle $C = v_1 e_1 v_2 e_2 \dots v_\ell e_\ell v_1$ in the $\Gamma$-labeled oriented graph $(G, \delta)$ we let $\lambda(C) = \lambda(e_1, v_1) \cdot \lambda(e_2, v_2) \cdot \ldots \cdot \lambda(e_\ell, v_\ell)$, and we call $C$ a *nonnull* cycle when $\lambda(C) \neq 1_\Gamma$. Now, in the Group FVS problem we are given a finite group $(\Gamma, \cdot)$ with unit element $1_\Gamma \in \Gamma$, a $\Gamma$-labeled oriented graph $(G = (V, E), \delta)$ with labeling function $\lambda$, an integer $k$, and the objective is to decide whether there exists a vertex set $X$ with $|X| \leq k$ such that there are no nonnull cycles in $G - X$.

It is well known that Group FVS generalizes Feedback Vertex Set, Odd Cycle Transversal, (Edge) Subset FVS, as well as Multiway Cut:

- For Feedback Vertex Set we can choose as $\Gamma$ the group $\mathbb{Z}_2^m$, where $m$ is the number of edges (cf. [IWY16]).
- For Odd Cycle Transversal we can choose as $\Gamma$ the cyclic group $\mathbb{Z}_2$ where each edge is labeled with the group element that is not the identity element (cf. [KW12]).
- For Edge Subset FVS we can choose as $\Gamma$ the group $\mathbb{Z}_2^{|S|}$ (cf. [CPP16]).
- For Multiway Cut we can choose as $\Gamma$ the group $\mathbb{Z}_2^h$, where $h$ is equal to $\lceil \log_2(|T| + 1) \rceil$ (cf. [KW12]).

Wahlström [Wah17] considered an even more general feedback problem which is based on so-called *biased graphs*. A *biased graph* is a pair $\Psi = (G = (V, E), \mathcal{B})$ of a graph $G$ and a set $\mathcal{B}$ of cycles in $G$ with the property that if two cycles $C, C' \in \mathcal{B}$ form a collection of three internally vertex-disjoint paths with shared endpoints then the third cycle of $C \cup C'$ is also contained in $\mathcal{B}$; the cycles in $\mathcal{B}$ are called *balanced cycles* of $G$, and we say that the cycle class $\mathcal{B}$ is *linear*. We call a cycle $C$ *unbalanced* when $C \notin \mathcal{B}$. Let $\mathcal{C}$ be the set of cycles that are not contained in $\mathcal{B}$, i.e., the set of unbalanced cycles. The cycle class $\mathcal{C}$ is called *co-linear*. It holds that if $C \in \mathcal{C}$ is an unbalanced cycle, and if $P$ is a path with endpoints in $C$ that is internally vertex-disjoint from cycle $C$ then at least one of the two cycles formed by $C \cup P$ that are not $C$ is also unbalanced, i.e., contained in $\mathcal{C}$. In the Biased Graph Cleaning problem (BGC), we are given a biased graph $\Psi = (G, \mathcal{B})$, an integer $k$, and the task is to decide whether there exists a set $X$ of at most $k$ vertices of $G$ such that $G - X$ contains only balanced cycles. To solve the Biased Graph Cleaning problem, Wahlström considers a local version of this problem, called Rooted Biased Graph Cleaning problem (RBGC), where given a biased graph $\Psi = (G, \mathcal{B})$ together with a *rooted vertex* $r \in V(G)$, and an integer $k$, the objective is to decide whether there exists a set $X$ of at most $k$ vertices such

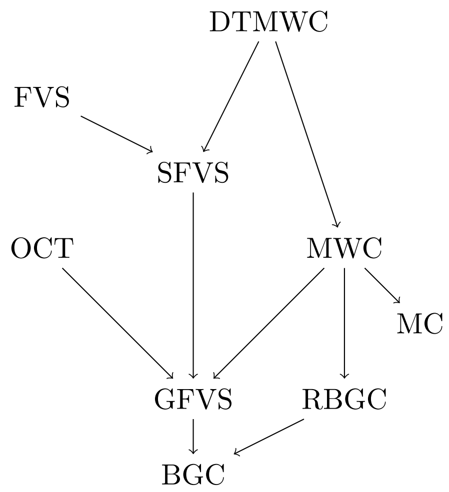



Figure 12.1.: An arrow from problem $P$ to problem $Q$ means that there exists a polynomial parameter transformation from $P$ parameterized by the solution size to $Q$ parameterized by the solution size.

that the connected component of $G - X$ that contains the root $r$ has only balanced cycles. Wahlström [Wah17] mentioned that Group FVS is a special case of Biased Graph Cleaning, because a $\Gamma$-labeled oriented graph is also a biased graph where every nonnull cycle is unbalanced and every other cycle is balanced. Additionally, Wahlström [Wah17] showed that one can reduce Multiway Cut to Rooted Biased Graph Cleaning: Given an instance $(G, T, k)$ of Multiway Cut one can construct an instance of Rooted Biased Graph Cleaning by duplicating each terminal $t \in T$ into $k+1$ copies, forming the set $T'$, and by adding a new vertex $r$ to $G$ that is the root vertex and that is adjacent to vertices in $T'$. Now, a cycle is unbalanced if and only if it contains the root $r$ and two vertices of the set $T'$ that are copies of two different terminals in $T$.

As for graph cut problems we can define Feedback Vertex Set as well as Subset FVS also for directed graphs. These problems are called Directed Feedback Vertex Set, short Directed FVS or DFVS, and Directed Subset Feedback Vertex Set, short Directed Subset FVS. In the Directed FVS problem we are given a directed graph $G$, an integer $k$, and the task is to decide whether there exists a vertex set $X$ with $|X| \leq k$ such that $G - X$ is acyclic, i.e., there is no directed cycle in $G - X$. Similarly, in the Directed Subset FVS problem we are given a directed graph $G$, an integer $k$, and additionally a vertex set $S \subseteq V(G)$, and the task is to find a set $X \subseteq V(G)$ such that $G - X$ contains no directed cycle $C$ that contains a vertex of $S$.

Some of the algorithms that solve feedback problems use a graph cut problem as a subroutine. For example, the fpt-algorithm of Reed et al. [RSV04] for Odd Cycle Transversal parameterized by the solution size $k$ computes certain vertex cuts (see

Section 12.2 for more details). Similarly, the fpt-algorithm of Cygan et al. [CPPW13b] for Edge Subset FVS parameterized by the solution size $k$ constructs and solves Multicut instances.

Moreover, the analysis of *graph cut* problems and *feedback* problems helped to develop new techniques in parameterized complexity for fpt-algorithms as well as kernelization algorithms. In terms of fpt-algorithms Reed et al. [RSV04] introduced the concept *iterative compression* (see also [GMN09]), Marx [Mar06] introduced *important separators*, and Marx and Razgon [MR14] introduced the technique of *random sampling of important separators* to obtain so-called *shadowless* solutions.[4] Besides techniques for fpt-algorithms, Kratsch and Wahlström [KW12, KW14] introduced matroid based tools to obtain (randomized) polynomial kernels.

## 12.2. Known Results

In this section we will summarize some known results for graph cut problems as well as feedback problems with respect to parameterized complexity.

**Feedback Vertex Set** The first fpt-algorithm for Feedback Vertex Set parameterized by the solution size $k$ is due to Downey and Fellows [DF92b] and runs in time $\mathcal{O}^*((2k+1)^k)$. After some improvements (cf. [Bod94, RSS06, KPS04]) Guo et al. [GGH$^+$06] gave the first single-exponential fpt-algorithm parameterized by the solution size $k$ with running time $\mathcal{O}^*(37.7^k)$. After some further improvements [CFL$^+$08, CCL15] Kociumaka and Pilipczuk [KP14] gave a deterministic fpt-algorithm that runs in time $\mathcal{O}^*(3.619^k)$. Each of the single-exponential fpt-algorithms uses iterative compression. Recently, Iwata and Kobayashi [IK19] gave a deterministic fpt-algorithm that runs in time $\mathcal{O}^*(3.46^k)$ by applying some reduction rules followed by highest-degree branching. Allowing randomness, Becker et al. [BBG00] showed that there exists a simple $\mathcal{O}^*(4^k)$ algorithm. This was improved by Cygan et al.[CNP$^+$11] to a randomized fpt-algorithm that runs in time $\mathcal{O}^*(3^k)$. To prove this, they first showed that Feedback Vertex Set can be solved in time $\mathcal{O}^*(3^{\mathrm{tw}})$, where tw denotes the treewidth of the input graph. Recently, Li and Nederlof [LN19] presented the currently best randomized fpt-algorithm which runs in time $\mathcal{O}^*(2.7^k)$.

Besides being fixed-parameter tractable, Feedback Vertex Set parameterized by the solution size $k$ also admits a polynomial kernel. The first polynomial kernel, due to Burrage et al. [BEF$^+$06], reduces to size $\mathcal{O}(k^{11})$. Bodlaender and Van Dijk [BvD10] improved this to a kernel of size $\mathcal{O}(k^3)$, and Thomassé [Tho10] further improved the kernel size to size $\mathcal{O}(k^2)$, more precisely to a graph with $4k^2$ vertices and $8k^2$ edges. The best known kernelization, due to Iwata [Iwa17], creates an equivalent instance with $2k^2+k$ vertices and $4k^2$ edges. This kernelization algorithm runs in linear time using the half-integral LP technique which was introduced by Iwata et al. [IWY16]. Furthermore,

[4] This final technique is sometimes called *shadow removal.*

Dell and Van Melkebeek [DvM14] showed that there exists no kernel of size $\mathcal{O}(k^{2-\varepsilon})$ for any constant $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

The Feedback Vertex Set problem has also been studied under different structural parameters. Jansen et al. [JRV14] showed, among other things, that Feedback Vertex Set parameterized by the size of a modulator to a chordal graph is fixed-parameter tractable [JRV14]. Since Bodlaender et al. [BJK14] showed that Feedback Vertex Set parameterized by the size of a modulator to a single clique has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, it follows that Feedback Vertex Set parameterized by the size of a modulator to a chordal graph does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Furthermore, Jansen et al. [JRV14] showed that Feedback Vertex Set parameterized by the size of a modulator to a mock-forest is fixed-parameter tractable but does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ and that Feedback Vertex Set parameterized by the size of a modulator to a pseudoforest admits a polynomial kernel with $\mathcal{O}(k^{10})$ vertices. Majumdar and Raman [MR18] improved the kernel for Feedback Vertex Set parameterized by the size of a modulator to a pseudoforest to $\mathcal{O}(k^6)$ vertices. Among other things, they also showed that Feedback Vertex Set parameterized by the size of a modulator to a mock-$d$-forest[5] admits a kernel with $O(k^{3d+3})$ vertices and that there exists no kernel of size $\mathcal{O}(k^{d+2-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

**Odd Cycle Transversal** Reed et al. [RSV04] showed that Odd Cycle Transversal can be solved in time $\mathcal{O}^*(4^k)$, where $k$ is the solution size, which implies that Odd Cycle Transversal is fixed-parameter tractable. The currently fastest fpt-algorithm for Odd Cycle Transversal parameterized by the solution size $k$ runs in time $\mathcal{O}^*(2.3146^k)$ [LNR$^+$14]. Kratsch and Wahlström [KW14] showed that Odd Cycle Transversal parameterized by the solution size $k$ admits a randomized polynomial kernel. This is the first kernelization algorithm that applies matroids.

The fpt-algorithm due to Reed et al. [RSV04] and the kernelization algorithm due to Kratsch and Wahlström [KW14] transform the Odd Cycle Transversal problem into a cut problem as follows (cf. [RSV04]): Let $(G, k)$ be an instance of Odd Cycle Transversal, and let $X$ be a vertex set such that $G - X$ is bipartite.[6] Note that we can assume that the set $X$ is an independent set in $G$; otherwise we can subdivide edges between vertices in $X$. Let $S_1$, $S_2$ be a bipartition of the bipartite graph $G - X$. We construct the graph $G' = (V', E')$ from $G = (V, E)$ and $X$, where $V' = (V \setminus X) \cup \{x_1, x_2 \mid x \in X\}$, and

$$E' = E[V \setminus X] \cup \{\{x_i, u\} \mid i \in \{1, 2\}, u \in S_{3-i}, x \in X, \{x, u\} \in E\}.$$

Observe, $G'$ is bipartite with bipartition $S_1 \cup \{x_1 \mid x \in X\}$ and $S_2 \cup \{x_2 \mid x \in X\}$. Reed et al. [RSV04] showed that the minimum size $Y \subseteq V$ such that $G - Y$ is bipartite

[5]A mock-$d$-forest is a mock-forest where each component has at most $d$ cycles.

[6]In the kernelization algorithm the set $X$ is an approximate solution. Since the fpt-algorithm uses iterative compression, we always consider instances of OCT where a solution is part of the input.

is equal to the minimum of $|X \setminus U|$ plus the size of an $(S, T)$-vertex cut in the graph $G' - \{x_i \mid x \in X \setminus U\}$ over all subsets $U$ of $X$ and all partitions $S$, $T$ of $\{x_i \mid x \in X \setminus U\}$ with $|S \cap \{x_1, x_2\}| = |T \cap \{x_1, x_2\}| = 1$.

**Subset FVS** Cygan et al. [CPPW13b] and Kawarabayashi and Kobayashi [KK12] independently showed that SUBSET FVS is fixed-parameter tractable. The fpt-algorithm of Cygan et al. runs in time $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$, while the one of Kawarabayashi and Kobayashi runs in time $\mathcal{O}(f(k) \cdot n^2 m)$, where $f$ is a function that depends super-exponentially on $k$. Wahlström [Wah14] then gave the first single-exponential algorithm with running time $4^k \cdot n^{\mathcal{O}(1)}$. An algorithm with subexponential dependence on $k$ is ruled out under the Exponential-Time Hypothesis (e.g., because SUBSET FVS generalizes VERTEX COVER). More recently, Lokshtanov et al. [LRS18] gave algorithms with deterministic time $2^{\mathcal{O}(k \log k)} \cdot (n + m)$ and randomized time $\mathcal{O}(25.6^k \cdot (n + m))$. We will show in Chapter 13 that SUBSET FVS parameterized by the solution size has a randomized polynomial kernel.

**Group FVS** Guillemot [Gui11] showed that GROUP FVS can be solved in time $\mathcal{O}^*((4|\Gamma| + 1)^k) = \mathcal{O}^*(2^{\mathcal{O}(k \cdot \log(|\Gamma|))})$ when $\Gamma$ is described by its multiplication table. This implies that GROUP FVS parameterized by $|\Gamma|$ and $k$ is fixed-parameter tractable. Assuming that the operations of the group $\Gamma$ are computed by a given black-box, Cygan et al. [CPP16] showed that GROUP FVS is solvable in time $\mathcal{O}^*(2^{\mathcal{O}(k \cdot \log(k))})$; hence GROUP FVS is also fixed-parameter tractable when parameterized only by the solution size $k$. Both fpt-algorithms use iterative compression and the algorithm of Cygan et al. [CPP16] reduces each problem of the compression step to a MULTIWAY CUT instance. Iwata et al. [IWY16] (see also [Wah14]) gave the first single exponential fpt-algorithms for GROUP FVS which has running time $\mathcal{O}^*(4^k)$. They also assumed that the group operations of $\Gamma$ are given via a black-box. Their algorithm uses LP-relaxations that extend the search space from $\{0, 1\}^{|V|}$ to $\{0, \frac{1}{2}, 1\}^{|V|}$, and has the benefit that the relaxed problem can be solved in polynomial time.

Using the matroid based tools, Kratsch and Wahlström [KW12] showed that GROUP FVS parameterized by the solution size admits a randomized polynomial kernel with $\mathcal{O}(k^{2s+2})$ vertices, where $s$ is the number of elements of the group $\Gamma$.

**(Rooted) Biased Graph Cleaning** Wahlström [Wah17] considered the BIASED GRAPH CLEANING and the ROOTED BIASED GRAPH CLEANING problem and showed that these problems can be solved in time $\mathcal{O}^*(4^k)$ and $\mathcal{O}^*(2^k)$, respectively. To obtain these results Wahlström [Wah17] shows that biased graphs have a half-integral LP-relaxation. This builds on an idea of Iwata et al. [IWY16] (see also [Wah14]). Iwata et al. [IWY16] showed how to use CSP tools to study and find half-integral relaxations of certain problems. The advantage of the biased graph framework is that it is a combinatorical condition that can be verified easily whereas, in general, it is not known how to decide whether there exists such a relaxation of a CSP (cf. [Wah17]).

**Directed FVS** For a long time it was an open problem whether Directed FVS is fixed-parameter tractable when parameterized by the solution size $k$ [DF92a]. In 2008, Chen et al. [CLL$^+$08] gave an $\mathcal{O}^*(4^k k!)$ algorithm for Directed FVS which proves that Directed FVS is fixed-parameter tractable when parameterized by $k$. One of the techniques they use is iterative compression. In each compression step their algorithm reduces instances of Directed FVS to certain instances of a multicut problem, which they call Skew Separator, where given a directed graph $D = (V, A)$, two collections $[S_1, S_2, \ldots, S_p]$ and $[T_1, T_2, \ldots, T_p]$ of $p$ vertex sets that are all pairwise disjoint and where the sets $S_1, S_2, \ldots, S_p$ have only outgoing edges and the sets $T_1, T_2, \ldots, T_p$ have only ingoing edges, an integer $k$, and the question is whether there exists a set $X \subseteq V \setminus \bigcup_{i=1}^{p}(S_i \cup T_i)$, with $|X| \leq k$, such that for all pairs $i$, $j$ of indices satisfying $1 \leq j \leq i \leq p$ it holds that there is no $S_i$,$T_j$-path in $G - X$. Afterwards, they show that Skew Separator parameterized by the solution size $k$ is fixed-parameter tractable by giving an $\mathcal{O}^*(4^k)$ time algorithm. They solve the Skew Separator by branching over important separators, but note that this term was introduced later. Overall, this results in the $\mathcal{O}^*(4^k k!)$ time algorithm for Directed FVS.

Lokshtanov et al. [LRS16] slightly improved this by showing that there exists an fpt-algorithm for Directed FVS with running time $\mathcal{O}(4^k k! k^5 \cdot (n+m))$, i.e., they improved the polynomial factor. An open question is whether there exists an $\mathcal{O}^*(2^{\mathcal{O}(k)})$, or even an $\mathcal{O}^*(2^{o(k \log(k))})$ time algorithm for Directed FVS (cf. [LRS16]). Note that it is well known that Directed FVS cannot be solved in time $\mathcal{O}^*(2^{o(k)})$ under the Exponential Time Hypothesis (ETH) [CFK$^+$15, DFL$^+$07].

Another open question is whether Directed FVS has a polynomial kernel when parameterized by the solution size $k$. However, Bergougnoux et al. [BEG$^+$17] showed that Directed FVS has a polynomial kernel with $\mathcal{O}(\ell^4)$ vertices when parameterized by the size $\ell$ of a feedback vertex set of the underlying undirected graph. This parameter is always at least as large as the solution size.

**Directed Subset FVS** Chitnis et al. [CCHM15] showed that even Directed Subset FVS parameterized by the solution size $k$ is fixed-parameter tractable. They gave an $\mathcal{O}^*(2^{O(k^3)})$ time algorithm for this problem. As techniques they used iterative compression as well as random sampling of important separators which was introduced by Marx and Razgon [MR14] and generalized to directed graphs by Chitnis et al. [CHM13]. Furthermore, they improved the technique of random sampling on directed graphs by, for example, improving the success probability.

**Multiway Cut** Robertson and Seymour [RS95, RS04] showed that Multiway Cut is fixed-parameter tractable when parameterized by the solution size. However, their proof is nonconstructive and only states the existence of such an algorithm (for more details see [CFK$^+$15, Section 6.3]).

In 2004, Marx [Mar06] gave the first constructive fpt-algorithm for Multiway Cut parameterized by the solution size. For this algorithm Marx introduced the notion of *important separators*: Simply put, given a graph $G$ and two disjoint vertex sets $X$ and $Y$, a vertex set $S \subseteq V(G) \setminus (X \cup Y)$ is called an important $(X, Y)$-separator, if there is no $X, Y$-path in $G - S$ and $S$ fulfills two additional properties (see Definition 12.2 for a proper definition). Marx [Mar06] showed that the number of important separators of size at most $k$ is upper bounded by $4^{k^2}$, and that one can enumerate all important separators in uniformly polynomial time. Furthermore, he showed that there exists an optimum solution to the MULTIWAY CUT instance $(G, T = \{t_1, \ldots, t_s\}, k)$ that contains an important $(\{t_1\}, T \setminus \{t_1\})$-separator. Now, the algorithm for MULTIWAY CUT parameterized by the solution size $k$ takes a terminal vertex $t \in T$, that is not separated from all other vertices in $T$, and branches over all important $(\{t\}, T \setminus \{t\})$-separators. In each branch we recursively solve the problem for $(G - S, T, k - |S|)$, where $S$ is the important separator of our branch. This results in an algorithm with running time $\mathcal{O}^*(4^{k^3})$.

This algorithm was improved by Chen et al. [CLL09b] by giving an $\mathcal{O}^*(4^k)$ time algorithm. They prove implicitly that one can upper bound the number of important separators by $\mathcal{O}(4^k)$ and as Marx [Mar06] they branch over the important separators.[7] It is well known that this bound is essentially tight up to polynomial factors, more precisely, there exists a family of graphs where the number of important separators of size at most $k$ is $\Omega^*(4^k)$.

Guillemot et al. [Gui11] also gave an $\mathcal{O}^*(4^k)$ time algorithm for MULTIWAY CUT, by considering a LP formulation of a generalization of the MULTIWAY CUT problem, called PATH TRANSVERSAL problem, and by showing that this LP has an optimum half-integral solution.

Cygan et al. [CPPW13a] considered MULTIWAY CUT parameterized by the above lower bound parameter $\ell = k - \textsc{lp}$, where $\textsc{lp}$ is the size of an optimum fractional solution to the multiway cut LP. They showed that there exists an fpt-algorithm that solves MULTIWAY CUT parameterized by $\ell$ in time $\mathcal{O}^*(4^{k-\textsc{lp}})$. To this end, they apply some reduction rules and use, among other things, that the LP-relaxation of the MULTIWAY CUT LP is half-integral which was shown by Garg et al. [GVY04]. If none of their reduction rules is applicable then each neighbor of a terminal vertex $t \in T$ has LP value $\frac{1}{2}$. Afterwards, they pick an arbitrary vertex $v \in N(T)$ and branch on it, i.e., in one branch they add $v$ to the multiway cut, decrease $k$ by one, and delete $v$ from $G$, and in the other branch they assume that $v$ is not in the multiway cut and contract the edge $\{t, v\}$. They showed that in both cases the parameter $\ell$ decreases by at least $\frac{1}{2}$. This leads to the desired $\mathcal{O}^*(4^{k-\textsc{lp}})$ time algorithm and proves that MULTIWAY CUT is fixed-parameter tractable when parameterized by $k - \textsc{lp}$. Besides, they showed that one can reduce, in polynomial time, to an equivalent instance with at most $2k$ terminals improving a result due to Razgon [Raz11] which reduces to at most $2k(k+1)$ terminals.

Furthermore, they point out that the fpt-algorithm for MULTIWAY CUT parameterized by $\ell = k - \textsc{lp}$ leads to an $\mathcal{O}^*(2^k)$ time algorithm for MULTIWAY CUT [CPPW13a]: If $2\textsc{lp} \leq k$ then it follows from the existence of a half-integral LP solution that there

[7] For more details see Section 12.3.

exists a solution of size at most $k$ which we can compute in polynomial time. Otherwise, it holds that $k \leq 2\textsc{lp}$ which implies that $k - \textsc{lp} \leq \frac{k}{2}$. Thus, the algorithm for Multiway Cut parameterized by $\ell$ runs in time $\mathcal{O}^*(4^\ell) = \mathcal{O}^*(2^k)$. For the Edge Multiway Cut problem, this running time was already shown by Xiao [Xia10c]. Cao et al. [CCF14] broke the $\mathcal{O}^*(2^k)$ barrier for the edge version of Multiway Cut, by showing that Edge Multiway Cut can be solved in time $\mathcal{O}^*(1.84^k)$.

It is not known whether Multiway Cut admits a polynomial kernel. However, Kratsch and Wahlström [KW12], showed that the easier problem Deletable Terminal Multiway Cut admits a randomized polynomial kernel with $\mathcal{O}(k^3)$ many vertices. Furthermore, they showed that Multiway Cut admits a randomized polynomial kernel when the number of terminals is constant. More precisely, they showed that Multiway Cut admits a randomized polynomial kernel with $\mathcal{O}(k^{s+1})$ vertices, where $s$ is the number of terminal vertices. Both kernelizations use the matroid based tools.

**Multicut** Using important separators Marx [Mar06] showed that there exists an algorithm for Multicut that runs in time $\mathcal{O}^*(2^{sk}4^{k^3})$, where $s$ is the number of terminals and $k$ is the solution size. Thus, Multicut is fixed-parameter tractable when parameterized by the solution size $k$ and the number of terminals $s$. This was improved by Xiao [Xia10c] to an algorithm with running time $\mathcal{O}^*((2s)^{s+k/2}) = \mathcal{O}^*(2^{\log(2s)(s+k/2)})$, and then by Guillemot [Gui11] who gave an $\mathcal{O}^*((8s)^k) = \mathcal{O}^*(2^{\log(8s)k})$ time algorithm.

Independently, Bousquet et al. [BDT18] and Marx and Razgon [MR14] showed that Multicut is still fixed-parameter tractable when only parameterized by the solution size $k$. Both fpt-algorithms use iterative compression, but the compression steps are different. Bousquet et al. [BDT18] use reduction rules to reduce the compression instance to a 2-SAT formula. This results in an $\mathcal{O}^*(k^{\mathcal{O}(k^6)})$ time algorithm. In contrast, Marx and Razgon [MR14] introduce the technique of random sampling of important separators to obtain so-called shadowless solutions.[8] They apply this technique in the compression step to reduce to an Almost 2-SAT instance. Their algorithm has running time $\mathcal{O}^*(2^{\mathcal{O}(k^3)})$.

Cygan et al. [CKP+14] showed that Multicut parameterized by the solution size does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. However, for a constant number $s$ of terminal pairs, Kratsch and Wahlström [KW12] showed that there exists a randomized polynomial kernel for Multicut with $\mathcal{O}(k^{\lceil\sqrt{2s}\rceil+1})$ vertices, using the matroid based tools.

**Directed Multiway Cut** Chitnis et al. [CHM13] extended the notion of shadowless solutions, introduced by Marx and Razgon [MR14], to directed graphs. By adapting the technique of random sampling of important separators, of Marx and Razgon [MR14],

[8] Given an instance $(G, T, W, k)$ of the Multicut Compression problem where $W$ is a multicut of $(G, T)$, the task is to find a multicut $S$ of $(G, T)$ of size at most $k$ such that $S \cap W = \emptyset$ and such that $S$ is also a multiway cut of $(G, W)$. The set $S$ is a *shadowless* solution if every vertex of $G \setminus S$ is reachable from a vertex in $W$, i.e., $G \setminus S$ has exactly $|W|$ connected components.

| Problem | FPT | Kernel |
|---|---|---|
| Feedback Vertex Set | $\mathcal{O}^*(3.619^k)$ | $4k^2$ |
| Odd Cycle Transversal | $\mathcal{O}^*(2.3146^k)$ | randomized |
| Subset FVS | $\mathcal{O}^*(4^k)$ | randomized[9] |
| Group FVS | $\mathcal{O}^*(4^k)$ | unknown, but $\mathcal{O}(k^{2s+2})$ |
| Biased Graph Cleaning | $\mathcal{O}^*(4^k)$ | unknown |
| Directed FVS | $\mathcal{O}^*(4^k k!)$ | unknown |
| Directed Subset FVS | $\mathcal{O}^*(2^{\mathcal{O}(k^3)})$ | unknown |
| Multiway Cut | $\mathcal{O}^*(2^k)$ | unknown, but $\mathcal{O}(k^{s+1})$ |
| Deletable Terminal MWC | $\mathcal{O}^*(2^k)$ | $\mathcal{O}(k^3)$ |
| Multicut | $\mathcal{O}^*(2^{\mathcal{O}(k^3)})$ | no kernel, but $\mathcal{O}(k^{\lceil\sqrt{2s}\rceil+1})$ |
| Directed Multiway Cut | $\mathcal{O}^*(2^{\mathcal{O}(k^2)})$ | no kernel |
| Directed Multicut | $W[1]$-hard | no kernel |

Table 12.1.: Results with parameter $k$ solution size. By $s$ we denote the number of group elements, the number of terminals or the number of terminal pairs, when considering Group FVS, MWC, or Directed MC, respectively.

to directed graphs to obtain shadowless solutions, Chitnis et al. [CHM13] showed that Directed Multiway Cut can be solved in time $\mathcal{O}^*(2^{2^{\mathcal{O}(k)}})$. Hence, Directed Multiway Cut is fixed-parameter tractable when parameterized by the solution size $k$. This algorithm was improved by Chitnis et al. [CCHM15] to run in time $\mathcal{O}^*(2^{\mathcal{O}(k^2)})$, by making the random sampling process more efficient.

Furthermore, it was shown by Cygan et al. [CKP+14] that Directed Multiway Cut, even when restricted to two terminals, does not have a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

**Directed Multicut** Marx and Razgon [MR14] showed that Directed Multicut parameterized by the solution size $k$ is $W[1]$-hard. However, the result that Directed Multiway Cut is fixed-parameter tractable when parameterized by the solution size $k$, implies that Directed Multicut is fixed-parameter tractable when restricted to instances with two terminal pairs because, as mentioned above, Directed Multiway Cut restricted to two terminals is equivalent to Directed Multicut restricted to two terminal pairs. Thus, one could ask whether Directed Multicut is fixed-parameter tractable when parameterized by the solution size $k$ and the number $s$ of terminals. Unfortunately, Pilipczuk and Wahlström [PW18] showed that Directed Multicut parameterized by the solution size $k$ is still $W[1]$-hard when restricted to instances with only four terminal pairs. Thus, the only open question is whether Directed Multicut parameterized by the solution size is fixed-parameter tractable when restricted to three terminal pairs.

[9] We show this in Chapter 13.

Kratsch et al. [KPPW15] considered a special case of Directed Multicut, namely Directed Multicut in directed acyclic graphs. Using the framework of random sampling of important separators, introduced by Marx and Razgon [MR14], which Chitnis et al. [CHM13] extended to directed graphs, they were able to show that Directed Multicut restricted to directed acyclic graphs is fixed-parameter tractable when parameterized by the solution size $k$ and the number of terminal pairs $s$ by proving that the problem can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(s^2k+s2^{\mathcal{O}(k)})})$. Furthermore, they showed that Directed Multicut remains $W[1]$-hard on directed acyclic graphs when parameterized only by the solution size.

Overall, for all of the above mentioned graph cut problems as well as feedback problems it is well known whether these problems are fixed-parameter tractable when parameterized by the solution size. For some problems there are also fpt-algorithms for structural parameters. In contrast, to prove that these problems have or do not have a polynomial kernel seems to be much harder. The only tool, that seems to be powerful enough to obtain a kernelization algorithm for these problems, except for Feedback Vertex Set, are the matroid based tools of Kratsch and Wahlström [KW12, KW14].

## 12.3. Important Separators and Matroids

Two of the most important techniques, that were developed for graph cut problems and feedback problems, are *important separators* for fpt-algorithms and the *matroid based tools* for kernelizations. So far, every kernelization algorithm for a graph cut respectively feedback problem, except Feedback Vertex Set, does use the matroid based tools of Kratsch and Wahlström [KW12, KW14]. Furthermore, important separators were evolved to work also for directed graphs [CHM13] and are the basis for many fpt-algorithms.

### 12.3.1. Important Separators

In this subsection we give an introduction to important separators. First of all, we will give a proper definition of (important) separators. Afterwards, we will show some well-known properties of important separators to finally describe an fpt-algorithm for Multiway Cut parameterized by the solution size $k$ with running time $\mathcal{O}^*(4^k)$ (cf. [CLL09b, CFK$^+$15]).

**Definition 12.1** (Reachable Sets)**.** Let $G = (V, E)$ be an undirected graph, let $X \subseteq V$ be a vertex set, and let $S \subseteq V \setminus X$ be a vertex set that is disjoint from $X$. By $R_G(X, S)$ we denote the vertices that are *reachable* from $X$ in $G - S$, i.e., the set of vertices $v \in V \setminus (X \cup S)$ such that there exists an $X, v$-path in $G - S$. We drop the subscript $G$ when it is clear from the context.

**Definition 12.2** (Important Separator (cf. [Mar06]))**.** Let $G = (V, E)$ be an undirected graph, and let $X, Y \subseteq V(G)$ be two disjoint vertex sets. A vertex set $S \subseteq V \setminus (X \cup Y)$

is called an $(X,Y)$*-separator* if there exists no $X,Y$-path in $G-S$, or in other words, if the set $R(X,S)\cap Y$ is empty. By $\lambda_G(X,Y)$ we denote the size of a smallest $(X,Y)$-separator in $G$. Again, we drop the subscript $G$ when it is clear from the context. An $(X,Y)$-separator $S'$ is said to *dominate* an $(X,Y)$-separator $S$ if $|S'|\le|S|$ and $R(X,S)\subsetneq R(X,S')$. We call an $(X,Y)$-separator $S$ *important* if it is inclusionwise minimal and if there exists no $(X,Y)$-separator $S'$ that dominates $S$.

First of all, we make two simple observations about the connection between (inclusionwise minimal) separators and the neighborhood of their reachable sets.

Observation 12.3. Let $G=(V,E)$ be an undirected graph, and let $X,Y\subseteq V(G)$ be two disjoint vertex sets.

(i) If $S$ is an inclusionwise minimal $(X,Y)$-separator, then $N(R(X,S))=S$.

(ii) Let $A\subseteq V\setminus Y$ be a superset of $X$. It holds that $R(X,N(A))\subseteq A$. Furthermore, if $A$ is connected then equality holds.

Now, we will show some basic and well-known properties of important separators (cf. Marx [Mar06]). These properties help us to prove that the number of important separators of size at most $k$ is bounded by $4^k$ and that we can enumerate them in uniformly polynomial time.

**Lemma 12.4 (cf. [Mar06]).** *Let $G=(V,E)$ be an undirected graph, and let $X,Y\subseteq V$ be two disjoint vertex sets.*

*(i) In polynomial time we can check whether a given $(X,Y)$-separator $S$ is important. Furthermore, if $S$ is not an important $(X,Y)$-separator then we can find in polynomial time an important $(X,Y)$-separator $\hat{S}$ of size at most $|S|$ such that $R(X,S)\subseteq R(X,\hat{S})$.*

*(ii) There exists exactly one important $(X,Y)$-separator $S^*$ of size $\lambda(X,Y)$, i.e., the minimum important $(X,Y)$-separator is unique. Furthermore, one can compute this minimum important $(X,Y)$-separator in polynomial time and for each important $(X,Y)$-separator $S$ it holds that $R(X,S^*)\subseteq R(X,S)$.*

**Proof.** (i) Let $S\subseteq V$ be an $(X,Y)$-separator. It is well known that we one can check in polynomial time whether a vertex set is an $(X,Y)$-separator or not. First, we try to figure out whether $S$ is minimal, by testing for each vertex $v\in S$ whether $S'=S\setminus\{v\}$ is an $(X,Y)$-separator. If this is the case for at least one vertex $v\in S$ then $S$ is not an important $(X,Y)$-separator, because $S$ is not minimal.

Second, we check for each vertex $v\in S$ whether there exists an $(X,Y)$-separator $S'$ of size at most $|S|$ such that $R(X,S)\cup\{v\}\subseteq R(X,S')$ by computing a minimum $(X',Y)$-separator in $G$ in polynomial time, where $X'=R(X,S)\cup\{v\}$. Again, if such an $(X',Y)$-separator $S'$ of size at most $|S|$ exists, than $S'$ dominates $S$ which implies that $S$ is not an important $(X, Y)$-separator. Otherwise, it follows from the definition of important separators that $S$ is an important $(X, Y)$-separator.

Observe that the above construction can be used to find an important $(X, Y)$-separator $\hat{S}$ of size at most $S$ such that $R(X, S) \subseteq R(X, \hat{S})$ in polynomial time. As long as the set $S$ is not an important $(X, Y)$-separator we compute an $(X, Y)$-separator $S'$ of size at most $|S|$ such that $S' \subsetneq S$ or $S'$ dominates $S$ and repeat with the set $S = S'$. Since in each iteration the set of vertices that are reachable from $X$ increases or the size of the separator decreases, we do this procedure at most $|V| + |S|$ many times.

(ii) First of all, it follows from item (i) that there exists at least one important $(X, Y)$-separator of size $\lambda(X, Y)$ and that we can compute an important $(X, Y)$-separator of size $\lambda(X, Y)$ in polynomial time.

Now, assume for contradiction that there exist two different minimum important $(X, Y)$-separator $S_1$ and $S_2$ of size $\lambda(X, Y)$. We will use the well-known fact that for two vertex sets $A, B$ it holds that $|N(A)|+|N(B)| \geq |N(A \cap B)|+|N(A \cup B)|$.[10]

Let $A = R(X, S_1)$, $B = R(X, S_2)$, $S_B = N(A \cap B)$ and $S_T = N(A \cup B)$. It follows that $S_B$ and $S_T$ are $(X, Y)$-separators in $G$ because $R(X, S_B) \subseteq A \cap B$ and $R(X, S_T) = A \cup B$ (Observation 12.3) and $X$ is contained in $R(X, S_B)$ and $R(X, S_T)$. Since $S_1$ and $S_2$ have size $\lambda(X, Y)$, since $S_1 = N(A)$ and $S_2 = N(B)$ (Observation 12.3), and since $|N(A)| + |N(B)| \geq |N(A \cap B)| + |N(A \cup B)|$ it holds that $S_B$ and $S_T$ are $(X, Y)$-separators of size $\lambda(X, Y)$. But, it holds that $R(X, S_i) \subsetneq R(X, S_T) = R(X, S_1) \cup R(X, S_2)$ which contradicts the assumption that $S_1$ and $S_2$ are important $(X, Y)$-separators because $S_T$ is an $(X, Y)$-separator that dominates $S_1$ and $S_2$.

It remains to show that $R(X, S^*) \subseteq R(X, S)$ for all important $(X, Y)$-separators in $G$, where $S^*$ is the unique minimum important $(X, Y)$-separator of $G$. As before, we can consider the two $(X, Y)$-separators $S_B = N(R(X, S^*) \cap R(X, S))$ and $S_T = N(R(X, S^*) \cup R(X, S))$ in $G$. Since $|S_B| \geq |S^*|$ it follows that $|S_T| \leq |S|$. But this implies that $S_T$ dominates the set $S$ if $R(X, S^*) \subsetneq R(X, S)$, because $R(X, S) \subseteq R(X, S_T) = R(X, S^*) \cup R(X, S)$. This concludes the proof. ■

Now, we show inductively that the number of important $(X, Y)$-separators of size at most $k$ in graph $G$, where $X$ and $Y$ are two disjoint vertex sets, is bounded by $2^{2k-\lambda(X,Y)}$. This implies the above mentioned upper bound of $4^k$ important separators.

**Lemma 12.5 (cf. [CLL09b, CFK$^+$15]).** *Let $G = (V, E)$ be an undirected graph, and let $X, Y \subseteq V(G)$ be two disjoint vertex sets. For every integer $k \geq 0$ there are at most $4^k$ important $(X, Y)$-separators of size at most $k$. Additionally, there exists an algorithm that enumerates all important $(X, Y)$-separators of size at most $k$ in time $\mathcal{O}^*(|\mathcal{S}_k|)$, where $\mathcal{S}_k$ is the set of important $(X, Y)$-separators of size at most $k$.*

[10]It is well known that $|N(A)| + |N(B)| = |N(A \cup B)| + |N(A) \cap N(B)| + |N(A) \cap B| + |N(B) \cap A| \geq |N(A \cap B)| + |N(A \cup B)|$.

**Proof.** We will prove a stronger result, namely that the number of important $(X, Y)$-separators of size at most $k$ is upper bounded by $2^{2k-\lambda(X,Y)} \le 4^k$, by induction over $2k-\lambda(X, Y)$. Obviously, if $k < \lambda(X, Y)$ then there exists no important $(X, Y)$-separator of size at most $k$ and $2^{2k-\lambda(X,Y)} < 2^{\lambda(X,Y)}$. Furthermore, if $\lambda(X, Y) = 0$ and $k \ge 0$ then the only important $(X, Y)$-separator is the empty set. Overall, this shows that the statement holds when $2k - \lambda(X, Y) < \lambda(X, Y)$ or when $\lambda(X, Y) = 0$.

Now, assume that $\lambda(X, Y) > 0$, and that $2k - \lambda(X, Y) \ge \lambda(X, Y)$. Let $S^*$ be the unique minimum important $(X, Y)$-separator, i.e., $|S^*| = \lambda(X, Y)$, and let $S$ be an important $(X, Y)$-separator of size at most $k$. Since $\lambda(X, Y) > 0$ there exists a vertex $v \in S^*$. This vertex $v$ is either contained in $S$ or not.

If the vertex $v$ is contained in the important $(X, Y)$-separator $S$ then $S \setminus \{v\}$ is an $(X, Y)$-separator in $G - v$ of size at most $k - 1$. We will show that $S \setminus \{v\}$ is also an important $(X, Y)$-separator in $G - v$. Assume for contradiction that $S \setminus \{v\}$ is not an important $(X, Y)$-separator in $G - v$. Thus, there exists an $(X, Y)$-separator $S'$ of size at most $|S| - 1$ in $G - v$ such that $S' \subsetneq S$ or $R(X, S) \subsetneq R(X, S')$. It holds that $S' \cup \{v\}$ is an $(X, Y)$-separator in $G$ of size at most $|S|$ with $S' \cup \{v\} \subsetneq S$ or $R_G(X, S) \subsetneq R_G(X, S' \cup \{v\})$. This contradicts the assumption that $S$ is an important $(X, Y)$-separator in $G$ and shows that $S \setminus \{v\}$ is also an important $(X, Y)$-separator in $G - v$. Observe that $\lambda_G(X, Y) = \lambda_{G-v}(X, Y) + 1$ because $v$ is contained in the $(X, Y)$-separator $S^*$ of size $\lambda_G(X, Y)$ in $G$ and because every $(X, Y)$-separator in $G - v$ of size $\lambda_{G-v}(X, Y)$ together with vertex $v$ is an $(X, Y)$-separator in $G$.

Overall, this shows that there exists a one-to-one correspondence between the important $(X, Y)$-separators in $G$ of size at most $k$ that contain vertex $v$ and the important $(X, Y)$-separators in $G - v$ of size at most $k - 1$. Since $2(k - 1) - \lambda_{G-v}(X, Y) = 2k - \lambda_G(X, Y) - 1$ it follows inductively that the number of important $(X, Y)$-separators of size at most $k$ that contain $v$ is bounded by $2^{2(k-1)-\lambda_G(X,Y)} = 2^{2k-\lambda_G(X,Y)-1}$.

If the vertex $v$ is not contained in the important separator $S$ then $S$ is also an $(X', Y)$-separator in $G$ with $X' = R(X, S^*) \cup \{v\}$, because $R(X, S^*) \subseteq R(X, S)$ and $v \in S^* \setminus S$. We will show by contradiction that $S$ is also an important $(X', Y)$-separator in $G$. Assume that $S$ is not an important $(X', Y)$-separator in $G$. This implies that there exists an $(X', Y)$-separator $S'$ of size at most $|S|$ such that $S' \subsetneq S$ or $R(X', S) \subsetneq R(X', S')$. But, $S'$ is also an $(X, Y)$-separator in $G$ because $R(X', S') = R(X, S')$ which contradicts the assumption that $S$ is an important $(X, Y)$-separator in $G$. Hence, $S$ is an important $(X', Y)$-separator in $G$. Observe that $\lambda_G(X, Y) < \lambda_G(X', Y)$; otherwise $S^*$ would not be an important $(X, Y)$-separator.

This shows that there is a one-to-one correspondence between the important $(X, Y)$-separators in $G$ of size at most $k$ that do not contain the vertex $v$ and the important $(X', Y)$-separators in $G$ of size at most $k$. It follows inductively that the number of important $(X, Y)$-separators of size at most $k$ that do not contain $v$ is bounded by $2^{2k-\lambda_G(X',Y)} \le 2^{2k-\lambda_G(X,Y)-1}$ because $\lambda_G(X, Y) < \lambda_G(X', Y)$.

Overall, this shows that the number of important $(X, Y)$-separators in $G$ is bounded by $2 \cdot 2^{2k-\lambda(X,Y)-1} = 2^{2k-\lambda(X,Y)}$. Furthermore, it follows from the construction together with Lemma 12.4 that we can enumerate all important $(X, Y)$-separators of size at most

$k$ in time $\mathcal{O}^*(4^k)$. However, it is possible to modify the above construction such that each leaf of the branching tree corresponds to an important $(X, Y)$-separator of size at most $k$, implying that we can enumerate the set $\mathcal{S}_k$ in time $\mathcal{O}^*(|\mathcal{S}_k|)$ time.[11] ■

Finally, we want to show how we can use important separators to obtain an fpt-algorithm for Multiway Cut parameterized by the solution size (cf. [Mar06, CLL09b]). The algorithm uses the fact that there exists a multiway cut that contains an important $(t, T \setminus \{t\})$-separator.

**Lemma 12.6 (cf. [Mar06]).** *Let $G = (V, E)$ be an undirected graph, let $T \subseteq V$ be a set of terminals, and let $t \in T$ be an arbitrary terminal vertex. If there exists a multiway cut $S$ in $G$ then there exists a multiway cut $S^*$ of size at most $|S|$ in $G$ that contains an important $(t, T \setminus \{t\})$-separator.*

**Proof.** Let $S_t = N(R(t, S)) \subseteq S$ be the set of vertices in $S$ that are adjacent to a vertex that is reachable from $t$ in $G - S$. Obviously, the set $S_t$ is a $(t, T \setminus \{t\})$-separator. It follows from Lemma 12.4 item (i) that there exists an important $(t, T \setminus \{t\})$-separator $S_t^*$ of size at most $|S_t|$ with $R(X, S_t) \subseteq R(X, S_t^*)$. We will show that $S^* = (S \setminus S_t) \cup S_t^*$ is also a multiway cut in $G$. Note that $S^*$ has size at most $|S|$ because $S_t \subseteq S$ and $|S_t^*| \leq |S_t|$. Since $S^*$ contains the $(t, T \setminus \{t\})$-separator $S_t^*$ it holds that there exists no $t, T \setminus \{t\}$-path in $G - S^*$. Thus, assume for contradiction that there exists a path $P$ between two terminal vertices $t_1, t_2 \in T \setminus \{t\}$ in $G - S^*$. The path $P$ must contain at least one vertex of the set $S$ because $S$ is a multiway cut in $G$. Thus, $P$ contains at least one vertex of the set $S \setminus S^* \subseteq S_t$ and no vertex of $S^*$. But, every vertex $v \in S_t$ is either contained in $S_t^*$ or in $R(t, S_t^*) \subseteq R(t, S^*)$ because $R(t, S_t) \subseteq R(t, S_t^*)$. This implies that $P$ contains a path from $t_1$ and $t_2$ in $G - S^*$ via a vertex in $R(t, S_t^*)$ which contradicts the assumption that $S_t^*$ is a $(t, T \setminus \{t\})$-separator. This shows that $S^*$ is a multiway cut and concludes the proof. ■

Finally, we use Lemma 12.5 and Lemma 12.6 to obtain an fpt-algorithm for Multiway Cut parameterized by the solution size $k$ by branching on important $(t, T \setminus \{t\})$-separators of size at most $k$.

**Theorem 12.7 (cf. [Mar06, CLL09b]).** Multiway Cut *parameterized by the solution size $k$ can be solved in time $\mathcal{O}^*(4^k)$.*

**Proof.** Let $(G, T, k)$ be an instance of Multiway Cut. If every terminal vertex of $T$ is contained in a single connected component of $G$ then we are done. Otherwise, we pick a terminal $t \in T$ that is not separated from at least one other terminal in $T$. Now, we branch on all important $(t, T \setminus \{t\})$-separators $S$ of size at most $k$. In each branch we solve the Multiway Cut instance $(G - S, T, k - |S|)$.

The correctness of the algorithm follows from Lemma 12.6 because if there exists a multiway cut of size at most $k$ in $G$ then there exists multiway cut of size at most $k$

[11] For more details see, for example, Cygan et al. [CFK+15, Thoerem 8.16].

in $G$ that contains an important $(t, T \setminus \{t\})$-separator for one terminal $t \in T$. Thus, $(G, T, k)$ is a yes-instance of Multiway Cut if and only if there exists an important $(t, T \setminus \{t\})$-separator such that $(G - S, T, k - |S|)$ is a yes-instance of Multiway Cut.

To prove that the algorithm runs in time $\mathcal{O}^*(4^k)$ we will prove, by induction over $k$, that the branching tree has at most $4^k$ leaves. For $k = 0$ the claim holds. Thus, for the induction hypothesis we assume that the claim is true for all parameter values less than $k$. This implies that every recursive call $(G - S, T, k - |S|)$ leads to a search tree with at most $4^{k-|S|}$ leaves. Let $\mathcal{S}$ be the set of important $(t, T \setminus \{t\})$-separators of size at most $k$. It holds that the search tree of instance $(G, T, k)$ has at most

$$\sum_{S \in \mathcal{S}} 4^{k-|S|} = 4^k \cdot \sum_{S \in \mathcal{S}} 4^{-|S|} \leq 4^k$$

leaves, where the last inequality holds because $\sum_{S \in \mathcal{S}} 4^{-|S|} \leq 2^{-\lambda}$, where we define $\lambda = \lambda(t, T \setminus \{t\})$: Obviously, for $k = \lambda$ it holds that $\sum_{S \in \mathcal{S}} 4^{-|S|} \leq 2^{-\lambda}$ because $\mathcal{S}$ contains only the unique minimum important $(T, T \setminus \{t\})$-separator $S^*$. Now, for $k > \lambda$, we choose a vertex $v \in S^*$ and distinguish whether $v$ is or is not contained in the important separator (as in the proof of Lemma 12.5). Let $\mathcal{S}_1 \subseteq \mathcal{S}$ be the set of important separators that contain $v$ and let $\mathcal{S}_2 \subseteq \mathcal{S}$ be the set of important separators that do not contain $v$. By induction hypothesis, it holds that $\sum_{S \in \mathcal{S}_1} 4^{-|S|} \leq 2^{-(\lambda-1)}$ because the set $\mathcal{S}_1$ is the set of important $(t, T \setminus \{t\})$-separators of size at most $k - 1$ in $G - v$ and the unique minimum important separator has size $\lambda - 1$. Furthermore, it also holds, by induction hypothesis, that $\sum_{S \in \mathcal{S}_2} 4^{-|S|} \leq 2^{-(\lambda+1)}$ because the set $\mathcal{S}_2$ is the set of important $(t, T \setminus \{t\})$-separators of size at most $k$ in $G/\{t, v\}$ and the unique minimum important separator has size at least $\lambda + 1$. Overall, this implies that

$$\sum_{S \in \mathcal{S}} 4^{-|S|} = \sum_{S \in \mathcal{S}_1} 4^{-(|S|+1)} + \sum_{S \in \mathcal{S}_2} 4^{-|S|} \leq 4^{-1} 2^{-\lambda+1} + 2^{-\lambda-1} \leq 2^{-\lambda}.$$

Since we can compute $\mathcal{S}$ in time $\mathcal{O}^*(|\mathcal{S}|)$ time, this shows that we can solve Multiway Cut in time $\mathcal{O}^*(4^k)$ time, which concludes the proof. ∎

### 12.3.2. Matroids

In this subsection we give an introduction to the matroid based tools for kernelization due to Kratsch and Wahlström [KW12, KW14]. We will use this tool to show in Chapter 13 that Subset FVS parameterized by the solution size admits a randomized polynomial kernel. To get an impression of the matroid based tools we will sketch a polynomial kernel for Odd Cycle Transversal parameterized by the solution size following Kratsch and Wahlström [KW12, KW14].

**Definition 12.8** (Matroid)**.** A *matroid* $M = (U, \mathcal{I})$ consists of a finite set $U$ and a family $\mathcal{I}$ of subsets of $U$, called *independent sets*, fulfilling the following properties:

(i) $\emptyset \in \mathcal{I}$,

(ii) if $X \subseteq Y$ and $Y \in \mathcal{I}$ then also $X \in \mathcal{I}$, and

(iii) if $X, Y \in \mathcal{I}$ with $|X| < |Y|$ then there exists $y \in Y \setminus X$ such that $X \cup \{y\} \in \mathcal{I}$.

The *rank* of a matroid $M$, denoted by $r(M)$, is the size of the largest independent set of the matroid $M$.

Let $A$ be a matrix over an arbitrary field $F$. Let $U$ be the set of columns of $A$ and let $\mathcal{I}$ be the family of all sets $X \subseteq U$ of columns that are linearly independent over $F$. Then $M = (U, \mathcal{I})$ is a matroid, called the *linear matroid* or *vector matroid* of $A$, and we say that $A$ *represents* $M$. Furthermore, we say that a matroid $M$ is *representable*, if there exists a matrix $A$ that represents $M$. If $M = (U, \mathcal{I})$ is representable over some field, then it is also representable by an $r(M) \times |U|$ matrix; by Gaussian elimination we can always reduce a representing matrix for $M$ to one with $r(M)$ many rows (cf. [Mar09]). Let $M_1 = (U_1, \mathcal{I}_1)$ and $M_2 = (U_2, \mathcal{I}_2)$ be two matroids with $U_1 \cap U_2 = \emptyset$. The *direct sum* $M_1 \oplus M_2$ is a matroid over $U = U_1 \cup U_2$ with independent sets $\mathcal{I} = \{X \subseteq U \mid X \cap U_1 \in \mathcal{I}_1, X \cap U_2 \in \mathcal{I}_2\}$. If $A_1$ and $A_2$ represent the two matroids over the same field $F$, then matrix $A = \mathrm{diag}(A_1, A_2) := \left(\begin{smallmatrix} A_1 & 0 \\ 0 & A_2 \end{smallmatrix}\right)$ represents $M_1 \oplus M_2$.

A special kind of matroid is the *uniform matroid* of rank $k$ over ground set $U$, where a set $X \subseteq U$ is independent if and only if $|X| \leq k$. It is well known that every uniform matroid is linear and can be represented by an $k \times |U|$ matrix. Moreover, the $k \times |U|$ Vandermonde matrix represents the uniform matroid of rank $k$ over ground set $U$.

Let $G = (V, E)$ be a graph that may have both directed and undirected edges and let $S \subseteq V$. A set $T \subseteq V$ is *linked* to $S$ if there exist $|T|$ vertex-disjoint paths from $S$ to $T$. Thus, every vertex in $T$ is the endpoint of a different path from $S$. It holds that $M = (U, \mathcal{I})$, where $U \subseteq V$ and $\mathcal{I}$ contains all sets $T \subseteq U$ that are linked to $S$ in $G$, is a matroid [Per68]. The matroid $M$ is also called the *gammoid* on $G$ with sources $S$ and ground set $U$; if $U = V$ then $M$ is also called a *strict gammoid.* Marx [Mar09] gave a randomized (Monte Carlo) polynomial-time procedure for finding a matrix representation of a strict gammoid. The error probability can be made exponentially small in the size of the graph. A matrix representation for a gammoid for graph $G = (V, E)$ with ground set $U \subsetneq V$ and sources $S$ can be obtained from the strict gammoid for $G$ and $S$ by simply deleting columns corresponding to elements of $V \setminus U$.

Let $A, B$ be independent sets in a matroid. We say that $A$ *extends* $B$ if $A \cap B = \emptyset$ and $A \cup B$ is again an independent set. Note, from the independence of $A \cup B$ the independence of $A$ and $B$ follows due to the second matroid property.

**Definition 12.9** (Representative Sets)**.** Let $M = (U, \mathcal{I})$ be a matroid, let $\mathcal{A} \subseteq \mathcal{I}$, and let $q \in \mathbb{N}$. A set $\mathcal{A}' \subseteq \mathcal{A}$ is *$q$-representative* for $\mathcal{A}$ if for every independent set $B$ of size at most $q$ there is a set $A \in \mathcal{A}$ that extends $B$ if and only if there is also a set $A' \in \mathcal{A}'$ that extends $B$.

Observe that if $\mathcal{A}'$ is $q$-representative for $\mathcal{A}$ and there exists a set $A \in \mathcal{A}$ that *uniquely extends* some given independent set $I$ of size at most $q$, then this implies that $A \in \mathcal{A}'$.

The following theorem of Lovász [Lov77] proves that for any linear matroid there exist small representative sets. It was made algorithmic by Marx [Mar09] and, thus, permits to find representative sets in polynomial time when given a matrix representation of the matroid. A faster algorithm for this task was developed by Fomin et al. [FLS14].

**Lemma 12.10 (Lovász [Lov77], Marx [Mar09]).** *Let $M$ be a linear matroid of rank $q+p$, and let $\mathcal{T} = \{I_1, I_2, \dots, I_t\}$ be a collection of independent sets of $M$, each of size $p$. If $|\mathcal{T}| > \binom{q+p}{p}$, then there is a set $I \in \mathcal{T}$ such that $\mathcal{T} \setminus \{I\}$ is $q$-representative for $\mathcal{T}$. Furthermore, given a representation $A$ of $M$, we can find such a set $I$ in $f(q,p) \cdot (\|A\|t)^{\mathcal{O}(1)}$ time, where $f(q,p)$ is a polynomial in $(p+q)^p$.*

Given a gammoid $M$ we can compute in randomized polynomial time a representation of the gammoid. Together with Theorem 12.10 it follows that given a gammoid $M$ and a collection $\mathcal{T} = \{I_1, \dots, I_t\}$ of independent sets, each of constant size $p$, we can find in randomized polynomial time a set $\mathcal{T}' \subseteq \mathcal{T}$ of size at most $\binom{q+p}{p}$ that is $q$-representative for $\mathcal{T}$.

To get an intuition how matroids (gammoids) and representative sets can be used to obtain randomized polynomial kernels, we discuss the polynomial kernel for Odd Cycle Transversal. As mentioned above, one can solve Odd Cycle Transversal by solving a terminal cut problem [RSV04]. Therefore, we will prove the following Theorem, due to Kratsch and Wahlström [KW12], which shows that there exists a sufficiently small set of vertices that is interesting for the cut problem.

**Theorem 12.11 (Kratsch and Wahlström [KW12]).** *Let $G = (V, E)$ be a directed graph, and let $X \subseteq V$ be a set of terminals. There exists a set $Z$ of $\mathcal{O}(|X|^3)$ vertices such that for any $S, T, R \subseteq X$, it holds that the set $Z$ contains a minimum $(S,T)$-vertex cut in $G - R$. We can find such a set $Z$ in randomized polynomial time with failure probability $\mathcal{O}(2^{|V|})$.*[12]

**Proof.** We call a vertex $v \in V$ *essential* if there exist sets $S, T, R \subseteq X$ such that $v$ is contained in every minimum $(S,T)$-vertex cut in $G - R$, and irrelevant otherwise. The kernelization algorithm will delete an irrelevant vertex as long as the graph contains more than $\mathcal{O}(|X|^3)$ many vertices.

Now, we construct a matroid $M$, consisting of three layers, to find all essential vertices. Let $M_0 = (V, \binom{V}{\leq |X|})$ be the uniform matroid of rank $|X|$ on $V$. For matroid $M_1$ we construct a graph $G_1$ from $G$ by adding for each vertex $v \in V \setminus X$ a sink-only copy, i.e., a parallel copy which retains only the ingoing edges, to $G$. Let $M_1$ be the gammoid on $G_1$ with sources $X$. Similar, for matroid $M_2$ we construct a graph $G_2$ from $G$ by first reversing all edges, and then we add for each vertex $v \in V \setminus X$ a sink-only copy to $G$. Let $M_2$ be the gammoid on $G_2$ with sources $X$. The matroid $M$ is the direct

[12]During the proof we compute a matrix that represents a gammoid. This can be done in randomized time (Monte Carlo) and leads to an one-side error probability, and only false positives.

sum of the matroids $M_0$, $M_1$ and $M_2$, i.e., $M = M_0 \oplus M_1 \oplus M_2$. Observe that $M$ is a linear matroid of rank $3|X|$, because $M_0$, $M_1$ and $M_2$ are linear matroids of rank $|X|$.

Next, we want to use representative sets to find the essential vertices. Therefore, we need a representation of the linear matroid $M$. As mentioned before, we can find such a representation in randomized polynomial time [Mar09]. This is the only part where we need randomization.

For a vertex $v \in V \setminus X$, we denote its copy in $M_0$ by $v(0)$, its two copies in $M_1$ by $v(1)$, $v'(1)$, and its two copies in $M_2$ by $v(2)$, $v'(2)$. Let $\mathcal{T} = \{(v(0), v'(1), v'(2)) \mid v \in V \setminus X\}$ and for each vertex $v \in V \setminus X$ let $T(v) = (v(0), v'(1), v'(2))$. We use Lemma 12.10 to find a $(3|X|-3)$-representative set $\mathcal{T}^* \subseteq \mathcal{T}$ for $\mathcal{T}$ in $M$. Let $V^* = \{v \in V \mid T(v) \in \mathcal{T}^*\}$.

We will show that every essential vertex is contained in the set $V^* \cup X$, by showing that for every vertex $v \in V \setminus X$, that is contained in every minimum $(S,T)$-vertex cut in $G-R$ for some sets $S, T, R \subseteq X$, it holds that the set $T(v)$ is contained in the set $\mathcal{T}^*$. Let $S, T, R \subseteq X$, and let $C$ be a minimum $(S,T)$-vertex cut in $G - R$. Let $v \in V \setminus X$ such that $v$ is contained in every minimum $(S,T)$-vertex cut in $G - R$, i.e., vertex $v$ is essential. It holds that $|C| < \min\{|S \setminus R|, |T \setminus R|\} \leq |X \setminus R|$ because $S \setminus R$ and $T \setminus R$ are $(S,T)$-vertex cuts in $G - R$ and because $v \in V \setminus X$ is contained in every minimum $(S,T)$-vertex cut.

Let $C'$ be the set that consists of the copy of $C - v$ in $M_0$, the copy of $C \cup R \cup (X \setminus S)$ in $M_1$ and the copy of $C \cup R \cup (X \setminus T)$ in $M_2$. The set $C'$ is an independent set of size at most $3|X| - 3$ in $M$: Since $M_0$ is the uniform matroid on $V$ of rank $|X|$ it holds that $C'$ restricted to matroid $M_0$ is an independent set of $M_0$ of size at most $|X| - 1$. Because $C$ is a minimum $(S,T)$-vertex cut in $G - R$, there are $|C|$ vertex-disjoint $(S \setminus R), C$-paths in $G - R$. Together with the $|(X \setminus S) \cup R|$ paths consisting of one vertex we obtain $|C \cup R \cup (X \setminus S)|$ vertex-disjoint paths from $X$ to $C \cup R \cup (X \setminus S)$ in $G$ and $G_1$. Hence, $C \cup R \cup (X \setminus S)$ is linked to $S$ in $G_1$ which implies that $C'$ restricted to $M_1$ is an independent set of $M_1$. It holds that $C'$ restricted to $M_1$ has size $|C \cup R \cup (X \setminus S)| = |C| + |R \cup (X \setminus S)| < |S \setminus R| + |R \cup (X \setminus S)| = |X|$. One can show similary that $C'$ restricted to $M_2$ is an independent set of $M_2$ of size at most $|X| - 1$. Overall, we showed that $C'$ is an independent set of matroid $M$ of size at most $3|X| - 3$.

Next, we will show that $T(v)$ uniquely extends the independent set $C'$ of $M$. Let $u \in V \setminus X$ such that $T(u)$ extends $C'$. It holds that vertex $u$ must be contained in $V \setminus C \cup \{v\}$, otherwise $T(u)$ intersects $C'$ restricted to $M_0$ which implies that $T(u)$ does not extend $C'$. In addition, vertex $u'(1) \in T(u)$ extends $C'$ restricted to $M_1$ if and only if there are $|C \cup R \cup (X \setminus S)| + 1$ many vertex-disjoint paths from $X$ to $C \cup R \cup (X \setminus S) \cup \{u'\}$ in $G_1$. Since $C$ is an $(S,T)$-vertex cut in $G - R$, since $u$ is not contained in $X$, and since $|C \cup R \cup (X \setminus S)|$ of these paths end in $C \cup R \cup (X \setminus S)$, it holds that $u'(1)$ is reachable from $S \setminus R$ in $G_1 - (C \cup R \cup (X \setminus S))$. Similary, one can show that vertex $u'(2)$ extends $C'$ restricted to $M_2$ if and only if $u'(2)$ is reachable from $T \setminus R$ in $G_2 - (C \cup R \cup (X \setminus T))$. Overall, this implies that $u$ is contained in $C$. Otherwise, there exists an $S,T$-path in $G - R - C$ because there exists a path from $S \setminus R$ to $u'(1)$ in $G_1 - C - R$ and therefore to $u$ in $G - C - R$, and because there exists a path from $u$ to $T \setminus R$ in $G - R - C$. Thus, $u = v$ is the only possible vertex that can extend $C'$.

It remains to show that $T(v)$ extends $C'$. Obviously, $v(0)$ extends $M_0$ by the choice of the set $C'$. Assume for contradiction that $v'(1)$ does not extend $C'$ restricted to $M_1$. This implies that $C'$ restricted to $M_1$ together with $u'(1)$ is dependent in the gammoid $M_1$. Thus, there are no $|C \cup R \cup (X \setminus S)| + 1$ vertex-disjoint paths from $X$ to $C \cup R \cup (X \setminus S) \cup \{v'(1)\}$ in $G_1$. It follows from the max-flow min-cut duality that there exists an $(X, C \cup R \cup (X \setminus S) \cup \{v'(1)\})$-vertex cut $C^*$ in $G_1$ of size at most $|C \cup R \cup (X \setminus S)|$. It holds that $v$ is not contained in $C^*$: Since $C \cup R \cup (X \setminus S)$ is an independent set of $M_1$ there are vertex-disjoint paths from $X$ to $C \cup R \cup (X \setminus S)$ which implies that $C^*$ intersects each of these paths exactly once. Thus, the sink-only copy $v'(1)$ is not contained in $C^*$. If vertex $v$ is contained in the minimum cut $C^*$, then it follows that there exists a path from $S$ to $v$ that avoids $C^* \setminus \{v\}$. However, this leads to an $S, v'(1)$-path in $G_1$ that avoids $C^*$ which contradicts the assumption that $C^*$ is an $(S, C \cup R \cup (X \setminus S) \cup \{v'(1)\})$-vertex cut in $G_1$. Thus, $v$ is not contained in $C^*$.

Now, $C^* \setminus (R \cup (X \setminus S))$ is also a minimum $(S,T)$-vertex cut in $G - R$ that does not contain $v$ which contradicts the assumption that $v$ is contained in every minimum $(S,T)$-vertex cut of $G - R$. Thus, $v'(1)$ extends $C'$ restricted to $M_1$. Similar, one can show that $v'(2)$ extends $C'$ restricted to $M_2$.

Overall, we showed that we can use representative sets to find in polynomial time a set $V^* \subseteq V \setminus X$ that contains all essential vertices. Therefore, we can make any vertex $v \in V \setminus (V^* \cup X)$ undeletable by removing $v$ from the graph and by adding an edge between each vertex in $N^-(v)$ and $N^+(v)$. Observe that this does not change the size of any $(S,T)$-vertex cut in $G - R$. Inductively, we obtain a set of $\mathcal{O}(|X|^3)$ vertices which covers at least one minimum $(S,T)$-vertex cut in $G - R$ for all choices of $S, T, R \subseteq X$. ∎

Now, we can use Theorem 12.11 to obtain a randomized polynomial kernel for Odd Cycle Transversal.

**Theorem 12.12 (Kratsch and Wahlström [KW12, KW14]).** Odd Cycle Transversal *admits a randomized polynomial kernel with $\mathcal{O}(k^{4.5})$ vertices.*

**Proof.** Let $(G, k)$ be an instance of Odd Cycle Transversal parameterized by the solution size $k$, with $G = (V, E)$. If $k \leq \log_2(|V|)$ then we run the algorithm of Reed et al. [RSV04], which runs in time $\mathcal{O}^*(4^k)$, which is polynomial when $k \leq \log_2(|V|)$. When the algorithm returns a solution of size at most $k$, we return a trivial yes-instance. Otherwise, we return a trivial no-instance.

Now, if $k \geq \log_2(|V|)$ then we run the polynomial-time $\mathcal{O}(\sqrt{\log_2(|V|)})$-factor approximation algorithm for Odd Cycle Transversal [ACMM05]. Since $\log_2(|V|) \leq k$ this algorithm returns a solution $X$ of size at most $\mathcal{O}(k^{\frac{1}{2}} \cdot \mathrm{OPT})$. If $|X| > \mathcal{O}(k^{1.5})$ then we return a trivial no-instance because $\mathrm{OPT} > k$. Thus, we assume that $X$ is an approximate solution of size $\mathcal{O}(k^{1.5})$.

We apply the construction of Reed et al.[RSV04] to obtain a graph cut problem. Let $S_1$ and $S_2$ be the two parts of the bipartite graph $G - X$. Let $G' = (V', E')$ be the bipartite graph with $V' = (V \setminus X) \cup \{x_1, x_2 \mid x \in X\}$ and $E' = E[V \setminus X] \cup \{\{x_i, u\} \mid i \in \{1,2\}, u \in S_{3-i}, x \in X, \{x, u\} \in E\}$. The two parts of $G'$ are $S_1 \cup \{x_1 \mid x \in X\}$ and $S_2 \cup \{x_2 \mid x \in X\}$. Let $U \subseteq X$, and let $S, T$ be a partition of $\{x_1, x_2 \mid x \in U\}$. We say that the pair $(S, T)$ is a *valid split* of $U$ if for all vertices $x \in U$ it holds that $|S \cap \{x_1, x_2\}| = |T \cap \{x_1, x_2\}|$.

Claim 12.13 (Reed et al. [RSV04]). *Let $Y \subseteq V$ such that $G - Y$ is bipartite with parts $A$ and $B$. The set $Y \setminus X$ is a minimum $(S, T)$-vertex cut in $G' - \{x_1, x_2 \mid x \in Y \cap X\}$, where $S = \{x_1 \mid x \in A\} \cup \{x_2 \mid x \in B\}$ and $T = \{x_1 \mid x \in B\} \cup \{x_2 \mid x \in A\}$.*

*Furthermore, let $U \subseteq X$, let $S, T$ be a partition of $\{x_1, x_2, \mid x \in X \setminus U\}$ such that $(S, T)$ is a valid split of $X \setminus U$, and let $C$ be an $(S, T)$-vertex cut in $G' - \{x_1, x_2 \mid x \in U\}$. The set $U \cup (C \setminus (S \cup T)) \cup \{x \in X \mid \{x_1, x_2\} \cap C \neq \emptyset\}$ is an odd cycle transversal of $G$.*

Thus, it follows from Clam 12.13 that we can apply Theorem 12.11 to graph $G'$ and vertex set $X_1 \cup X_2$ to obtain a set $Z$ of size $\mathcal{O}(|X|^3)$ that contains a minimum odd cycle transversal of $G$. It remains to shrink the graph correctly. For every pair of vertices $x, y \in X$, we add an edge or a path of length 2 between the vertices $x$ and $y$ if there exists a path of odd respectively even length between $x$ and $y$ whose internal vertices are disjoint from $X$. This concludes the proof. ■

There are only a few other kernelizations that use the matroid based tools, e.g., Vertex Cover parameterized by $k -$ lp [KW12] or by $k - (2$lp $-$ mm$)$ [Kra18] and Deletable Terminal Multiway Cut, $s$-Multiway Cut as well as Almost 2-SAT parameterized by the solution size [KW12]. We will show in the next Chapter that we can use the matroid based tools to obtain a randomized polynomial kernel for Subset FVS parameterized by the solution size.

# CHAPTER 13
# SUBSET FEEDBACK VERTEX SET

## 13.1. Introduction

Recall, in the Subset Feedback Vertex Set (Subset FVS) problem we are given an undirected graph $G = (V, E)$, a set of vertices $S \subseteq V$, an integer $k$, and we have to determine whether there is a set $X$ of at most $k$ vertices that intersects all cycles that contain at least one vertex of $S$. Clearly, because we can choose $S = V$, this is a generalization of the well-studied Feedback Vertex Set problem. As mentioned in the previous chapter, Cygan et al. [CPPW13b] and Kawarabayashi and Kobayashi [KK12] independently showed that Subset FVS is fixed-parameter tractable. Cygan et al. [CPPW13b] ask whether the Subset FVS problem also admits a polynomial kernelization and suggest that the matroid based tools of Kratsch and Wahlström [KW12], which we introduced in Section 12.3.2, could be applicable.

Interestingly, Cygan et al. [CPPW13b] claim a polynomial-time reduction from Multiway Cut to Subset FVS that does not change the parameter value and, hence, implies that Subset FVS is at least as hard as Multiway Cut regarding existence of polynomial kernels.[1] However, their construction [CPPW13b, Section 5] works only as a reduction of Deletable Terminal Multiway Cut to Subset FVS because it is not guaranteed that from a solution to the Subset FVS instance one obtains a solution with $k$ deleted *non-terminal* vertices. Since there is presently no such reduction from Multiway Cut to Subset FVS, a polynomial kernel for the latter does not imply one for the former. Thus, it makes sense to study kernels for Subset FVS before Multiway Cut.

**Our work.** Following joint work with Stefan Kratsch [HK18], we apply the matroid based tools of Kratsch and Wahlström [KW12] and we develop a randomized poly-

[1] The reduction builds upon a previous one by Even et al. [ENSZ00] who reduce Multiway Cut to Weighted Subset FVS using a single vertex in $S$ that has infinite weight.

nomial kernelization that reduces instances $(G, S, k)$ of Subset FVS to equivalent instances with at most $\mathcal{O}(k^9)$ vertices. This is our main result. Similarly to Cygan et al. [CPPW13b] we also work on Edge Subset FVS where $S$ is a set of edges of $G$ and $X$ needs to intersect all cycles that contain at least one edge of $S$. Since there are polynomial-time reductions between these two problems that do not change the solution size $k$, they are equivalent when it comes to kernelization complexity [CPPW13b]. The result is obtained in two parts.

First, we establish a randomized polynomial kernelization for Edge Subset FVS parameterized by $|S| + k$ that reduces to equivalent instances with at most $\mathcal{O}(|S|^2 k)$ vertices (Section 13.3). Note that nontrivial instances have $k < |S|$ since one could otherwise remove $S$ by deleting one endpoint of each edge in $S$. Thus, parameterization by $|S|$ suffices, but $\mathcal{O}(|S|^2 k)$ gives a better overall bound than $\mathcal{O}(|S|^3)$.

At a high level, this part is similar to the polynomial kernel for Deletable Terminal Multiway Cut. We show that certain solutions $X$, later called *dominant* solutions, allow particular path packings in the underlying graph $G$. For Deletable Terminal Multiway Cut this is achieved by a fairly simple replacement argument for solutions $X$ that are not sufficiently well-connected to connected components of $G - X$. For Edge Subset FVS the endpoints $T = V(S)$ of edges in $S$ can be regarded as terminals, but this gives a different separation property: A solution $X$ of Deletable Terminal Multiway Cut generates many connected components, because every connected component in $G - X$ contains at most one terminal. Whereas a solution $X$ of Edge Subset FVS does not need to generate many connected components in $G - X$ since only $S$-cycles need to be prevented, and thus connected components of $G - X$ may contain many vertices of $T$. Rather, in $G - X$ there must be a tree-like (or forest-like) structure with components without $S$-edges playing the role of nodes and with edges given by $S$. Nevertheless, using the tree-like structure, a replacement argument can be found, implying that dominant solutions must create many components in $(G - X) - S$ containing vertices of $T$ and be well-connected to them. This allows to set up a gammoid on $G - S$ with sources $T$ and apply, as in [KW12], a result of Lovász [Lov77] (made algorithmic by Marx [Mar09]) on representative sets in (linear) matroids that is then guaranteed to generate a superset of $X$. Randomization is only needed to generate a matrix representation for the gammoid.

In the second part (Section 13.4) we give a (deterministic) polynomial-time preprocessing that, given an instance $(G, S, k)$ of Edge Subset FVS, returns an equivalent instance $(G', S', k')$ with $k' \leq k$ and $|S'| \in \mathcal{O}(k^4)$. Together with the randomized kernelization from the first part this implies the claimed randomized kernelization to $\mathcal{O}(k^9)$ vertices.

A reduction of the number of $S$-edges is also a crucial ingredient in the fpt-algorithm for Edge Subset FVS by Cygan et al. [CPPW13b]. They achieve $|S| \in \mathcal{O}(k^3)$, but it is in a slightly more favorable setting: Using iterative compression, it suffices to solve the task of finding a solution $X'$ of size $k$ when given a solution $X$ of size $k + 1$. (This is well known in parameterized complexity, and we prefer not to repeat it here [RSV04].) Considering some unknown solution $X'$ of size $k$, one can guess the intersection $D$ of

$X'$ with $X$, by trying all $\mathcal{O}(2^{k+1})$ possibilities. For the correct guess $D = X' \cap X$, the remaining problem is to find for $(G - D, S \setminus D, k - |D|)$ a solution $Z'$ of size at most $k - |D|$ that is disjoint from $Z = X \setminus D$, since $Z' = X' \setminus D$ would be such a solution; here $S \setminus D$ denotes the set of edges in $S$ with no endpoint in $D$. Cygan et al. make the nice observation that the guessing also allows to assume that there is no other solution $X'$ with an even larger intersection with $X$.

In contrast, we cannot afford to run iterative compression for a kernelization to get a starting solution of size $k+1$ and, as is common, we have to start with an approximate solution $Z$, which can be assumed to be of size at most $8k$ using an 8-approximation algorithm of Even et al. [ENZ00]. The idea of guessing the intersection of an optimal solution with $Z$ is infeasible regarding both time and the number of created instances. Thus, while several structures like $z$-flowers or disjoint $x$,$y$-paths containing $S$-edges appear in both approaches, many things have to be handled differently. For example, having $k+2$ disjoint $x$,$y$-paths containing $S$-edges for $x, y \in Z$ implies that one of $x$ and $y$ must be in every solution of size $k$. Cygan et al. can stop here because the solution would not be disjoint from $Z$, whereas we need to instead store the information about $x$ and $y$ to later detect $S$-edges that can be safely removed. Like Cygan et al., we also use Gallai's $A$-path Theorem but we avoid the 2-expansion lemma by using the properties of a set of size at most $2k$, that intersects all cycles that contain at least on vertex of $S$ and contain a certain vertex, differently. (Such a set can be found if certain flowers of order $k+1$ do not exist, using Gallai's $A$-path Theorem.) Cygan et al. compute such a set $B$ of size at most $3k$ to find an $F$-flower of order $|X|$ (with $F \subseteq V$ outer-abundant; see [CPPW13b, Definition 3.4]) under the assumption that certain $F$-flowers of order $k+1$ do not exist and they show that there exists a solution that contains $X$ (under the assumption that there exists a solution that is disjoint from $F$). We cannot assume that our solution is disjoint from $F$ and we have to take another approach. Moreover, we observe that $z$-flowers can be found via matroid parity on an appropriate gammoid.[2]

## 13.2. Preliminaries

The following theorem about $A$-paths, where $A$ is a vertex set, was already used by Cygan et al. [CPPW13b] for Subset FVS and in the quadratic kernelization for Feedback Vertex Set by Thomassé [Tho10].

**Theorem 13.1 (Gallai [Gal61]).** *Let $A \subseteq V$ and $k \in \mathbb{N}$. If the maximum number of vertex-disjoint $A$-paths is strictly less than $k+1$, then there exists a set $B \subseteq V$ of at most $2k$ vertices that intersect every $A$-path.*

In particular it is possible to find either $(k+1)$-disjoint $A$-paths or a set $B$ that intersects all $A$-paths in polynomial time. This follows from Schrijver's proof of Gallai's theorem [Sch01].

[2]The latter is deterministic by applying a specialized matroid parity algorithm due to Tong et al. [TLV82].

Let $(G, S, k)$ be an instance of the EDGE SUBSET FVS problem. We call a cycle $C$ an *$S$-cycle*, if at least one edge of $S$ is contained in $C$. Let $x$ be a vertex of $V$. A set $\{C_1, C_2, \ldots, C_t\}$ of $S$-cycles that contain $x$ is called an *$x$-flower* of order $t$, if the sets of vertices $C_i \setminus \{x\}$ are pairwise disjoint. Note that if there exists an $x$-flower of order at least $k+1$, then the vertex $x$ must be in every solution for $(G, S, k)$, if one exists. A set $B \subseteq V \setminus \{x\}$ of size $t$ is called an *$x$-blocker* of size $t$, if each $S$-cycle through $x$ also contains at least one vertex of the set $B$.

The polynomial kernelization obtained in this chapter is randomized, which means that there is a small chance for the reduced instance to not be equivalent to the input. The error probability can be made exponentially small in the input size without increasing the size of the kernelization. Similarly to previous work [KW12], the only source for error is the need to compute a matrix representation for a gammoid.

## 13.3. Randomized Polynomial Kernelization for Parameter $|S| + k$

In this section we present a randomized polynomial kernelization for EDGE SUBSET FVS parameterized by $|S| + k$. Because deletion of one endpoint of each edge in $S$ always constitutes a feasible solution, nontrivial instances have $|S| > k$. Thus, our kernelization also works for parameter $|S|$ alone. However, to achieve a better bound for EDGE SUBSET FVS parameterized by $k$ only, it is beneficial to give the kernel size in terms of $|S|$ and $k$ rather than $|S|$ alone.

We use representative sets of independent sets of matroids to obtain a kernel of size $\mathcal{O}(|S|^2 k)$. Our approach is similar to the kernelization of DELETABLE TERMINAL MULTIWAY CUT($k$) [KW12]. As in that paper we construct path packings such that certain vertices can be shown to be in a representative set. Note that, unlike for multiway cut-type problems, a solution $X \subseteq V$ will not necessarily create many connected components. Rather, as used also in the fpt-algorithm of Cygan et al. [CPPW13b], it creates a particular tree-like structure in $G - X$. Nevertheless, endpoints of edges in $S$, denoted $T := V(S)$, will play the role of terminals that need to be separated in a certain way; hence a vertex $x$ in $T$ is called a *terminal.* We will focus on the graph $G - S$, i.e., with edges of $S$ deleted, in which a solution $X$ creates a grouping of (not deleted) terminals into connected components. The structure of these components will be crucial for a replacement argument (Lemma 13.3) that leads to the required path packing; this constitutes one of the key arguments for our result.

The kernelization consists of four steps. In the first step we show that if an instance is a yes-instance then there exists a solution $X$ with a certain path packing from $T$ to $X$. Then we define an appropriate gammoid to find in a next step a representative set of size $\mathcal{O}(|S|^2 k)$ which is (essentially) a superset of $X$ using Lemma 12.10. Finally we explain how to reduce the graph $G$, using the superset of the last step, to obtain an equivalent instance of EDGE SUBSET FVS.

Figure 13.1.: - - - $S$-edges. The vertices marked with a blue rectangle are an optimum subset feedback vertex set of the graph, but only the marked vertices in the left graph are a dominant solution.

**Analyzing solutions.** Let $(G, S, k)$ be a yes-instance of Edge Subset FVS. We say that a solution $X$ for $(G, S, k)$ is *dominant*, if it has minimum size and contains a maximal number of vertices from $T$ among solutions of minimum size. (See Figure 13.1 for an example.) The vertices in $X \cap T$ correspond to endpoints of edges in $S$ that we delete and the vertices in $X_0 = X \setminus T$ block all $x, y$-paths with $\{x, y\} \in S_0 = \{e \in S \mid e \cap X = \emptyset\}$, except the one that consists of the edge $\{x, y\}$. We show that $X$ is linked to $T$ in a strong sense, with vertices of $X_0$ playing a special role.

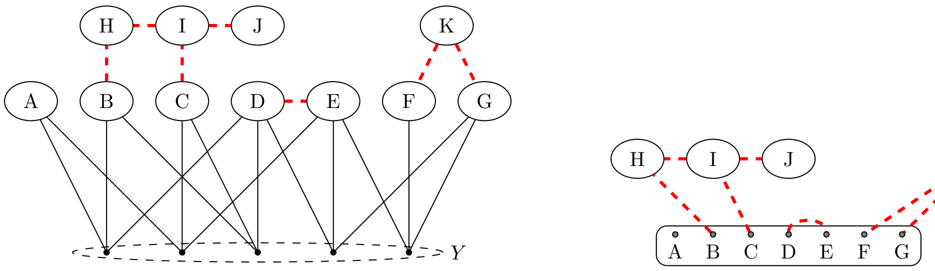
**Lemma 13.2.** *Let $X$ be a dominant solution for $(G, S, k)$ and let $x$ be any vertex in the set $X_0 = X \setminus T$. There exist $|X|+2$ paths from $T$ to $X$ in $G-S$ that are vertex-disjoint except for three paths ending in vertex $x$. Moreover, the paths can be chosen in such a way that each connected component of $G - X - S$ is intersected by at most one path.*

We will use Hall's Theorem (Theorem 2.1) and the lemma below to prove Lemma 13.2. For this purpose we consider the two graphs $G - X$ and $G - X - S$ which simplifies the analysis of a dominant solution. We call a connected component $K$ of $G - X - S$ *interesting* if it contains a terminal, i.e., if $T \cap V(K) = (T \setminus X) \cap V(K) \neq \emptyset$, and we say that a vertex $x \in X_0$ *sees* a connected component $K$ if $x$ is adjacent to a vertex of $K$ in $G$. We extend this definition by saying that a set $Y \subseteq X_0$ sees a component $K$ if at least one vertex $y \in Y$ sees $K$.

**Lemma 13.3.** *Let $X$ be a dominant solution and let $X_0 = X \setminus V(S)$. Every non-empty set $Y \subseteq X_0$ sees at least $|Y| + 2$ interesting components of $G - X - S$.*

**Proof.** Assume for contradiction that there exists a non-empty set $Y \subseteq X_0$ that sees at most $|Y| + 1$ interesting components of $G - X - S$. Let $\mathcal{C}$ denote the set of connected components of $G - X - S$, let $\mathcal{C}_i \subseteq \mathcal{C}$ denote the set of interesting components seen by $Y$, and let $\mathcal{C}_o \subseteq \mathcal{C}$ denote the other (non-interesting) components seen by $Y$. We will show that there is an alternative solution $X' = (X \setminus Y) \cup Y'$ that is smaller than $X$ or that contains more vertices of $T$, contradicting the choice of $X$ as a dominant solution. To construct $X'$ or, more precisely, to construct $Y'$ we consider the graphs $G - X$ and $G - (X \setminus Y)$.

We study the structure of $G - (X \setminus Y)$ to find a set $Y'$ that intersects all $S$-cycles in $G - (X \setminus Y)$. Accordingly, we are interested in the structure that is induced by the




(a) $\mathcal{C}_i^* = \{B, C, D, E, F, G\}$, $\mathcal{C}_o^* = \{A\}$.

(b) Graph $F'$.

Figure 13.2.: - - - $S$-edges. In the left graph, the ellipse-shaped bubbles represent the connected components of $G - X - S$ that are contained in a connected component of $C^+$ of $G - (X \setminus Y)$. Furthermore, the graph without the set $Y$ is a subgraph of $H_X$. The right graph is a subgraph of $H_{X \setminus Y}$.

$S$-edges in $G - (X \setminus Y)$ and want to compare it to the structure that is induced by the $S$-edges in $G - X$.

To this end, we define, for any subgraph $G - Z$ of $G$ with $Z \subseteq V$, the *$S$-component graph* $H_Z$ that has one vertex for each connected component of $G - Z - S$ and for every $S$-edge $e = \{x, y\} \in S$ with $\{x, y\} \cap Z = \emptyset$ the graph $H_Z$ has an edge between the two (not necessarily different) vertices which correspond to the connected components that contain at least one of the endpoints $x, y$ of $e$. Note that graph $H_Z$ can have parallel edges and loops. We say that the graph $G - Z$ is an *$S$-forest* if the $S$-component graph $H_Z$ is a forest. Obviously, it holds that a set $Z \subseteq V(G)$ is a solution of $G$ if and only if $G - Z$ is an $S$-forest. Observe that vertices that correspond to connected components without terminals in $G - Z$ are isolated in $H_Z$ because they are not incident with $S$-edges. Hence, non-interesting components of $G - X - S$ are isolated in the $S$-component graph $H_X$.

Now, we compare the $S$-component graphs of $G - X$ and $G - (X \setminus Y)$ to construct an alternative solution $X'$. Let $\mathcal{C}'$ denote the set of connected components of the graph $G - (X \setminus Y) - S$. The connected components of $G - X - S$ that are seen by $Y$ are not connected components in $G - (X \setminus Y)$, because we do not delete $Y$. Therefore, a connected component in $\mathcal{C}'$ either contains some vertices of $Y$ and some components in $\mathcal{C}_i \cup \mathcal{C}_o$ or is equal to a component in $\mathcal{C} \setminus (\mathcal{C}_i \cup \mathcal{C}_o)$. Thus, the consequence of not deleting $Y$ is that we merge some connected components in $\mathcal{C}_i \cup \mathcal{C}_o$. However, the $S$-component graphs $H_X$ and $H_{X \setminus Y}$ have the same number of edges, because the sets $X$ and $X \setminus Y$ intersect the same set of $S$-edges. This follows from the fact that $Y \subseteq X_0 = X \setminus T$ and hence that there are no additional vertices of $T$, i.e., $T \setminus X = T \setminus (X \setminus Y)$. In general, $G - (X \setminus Y)$ will not be an $S$-forest: The merging of vertices may lead to loops (from $S$-edges with both ends in the same component) and longer cycles in $H_{X \setminus Y}$.

We will show that deleting at most $|Y|$ edges of $S$, i.e., deleting a set $Y'$ of at most $|Y|$ endpoints of $S$-edges, will suffice for $G - ((X \setminus Y) \cup Y')$ to be an $S$-forest, making $(X \setminus Y) \cup Y'$ a valid solution.

Let $C^+$ be an arbitrary connected component in $G - (X \setminus Y)$ whose corresponding connected component in $H_{X \setminus Y}$ is not cycle-free. Note that $C^+$ is a union of connected components in $\mathcal{C}'$ that are connected by $S$-edges. Therefore $C^+$ must contain connected components in $\mathcal{C}$ that are seen by $Y$. Let $\mathcal{C}_i^* = \{C \in \mathcal{C}_i \mid C \subseteq C^+\}$ be the set of interesting components seen by $Y$ that are contained in $C^+$, let $\mathcal{C}_o^* = \{C \in \mathcal{C}_o \mid C \subseteq C^+\}$ be the set of non-interesting components seen by $Y$ that are contained in $C^+$, let $a = |\mathcal{C}_i^*|$, and let $b = |\mathcal{C}_o^*|$.

In $G - X$ the connected component $C^+$ may decompose into several separate connected components because we additionally delete the vertices of $Y$. Since $Y$ sees only components in $\mathcal{C}_i \cup \mathcal{C}_o$ the set $C^+$ decomposes into at most $a+b$ separate components by deleting $Y$. Recall that components in $\mathcal{C}_o$ are isolated in $G - X$ and contain no vertices of $T$ and, thus, they do not contribute any $S$-edges to $C^+$. It remains to consider the $a$ components of $\mathcal{C}_i^*$.

Every connected component in $\mathcal{C}_i^*$ corresponds to a vertex in the forest $H_X$. Let $F$ be the subforest of $H_X$ that contains all connected components of $H_X$ that contain at least one connected component of $\mathcal{C}_i^*$ as a vertex. Not deleting $Y$ corresponds to merging $a$ vertices in this forest into $d \geq 1$ new vertices. Let $F'$ be the connected subgraph in $H_{X \setminus Y}$ that results from $F$ by this operation; see Figure 13.2 for an illustration. If the subforest $F$ consists of $c$ vertices and, thus, at most $c-1$ $S$-edges then we obtain $c-a+d$ vertices that are connected by at most $c-1$ edges for the subforest $F'$. It therefore suffices to delete at most $(c-1) - ((c-a+d)-1) = a-d \leq a-1$ $S$-edges, i.e., to delete one endpoint of each of at most $a-1$ $S$-edges, to obtain a forest-structure in $F'$. Observe that we cannot delete just *any* $a-1$ edges but we can keep any $c-a+d-1$ $S$-edges spanning the $c-a+d$ components and delete the at most $a-1$ remaining $S$-edges.

Overall, we get that a connected component $C^+$ in $G - (X \setminus Y)$ that fully contains $a$ interesting components from $\mathcal{C}_i$ requires at most $a-1$ vertex deletions of endpoints of $S$-edges to obtain an $S$-forest. Since $Y$ sees at most $|Y|+1$ such components, the worst case is reached by a single component $C^+$ containing all $|Y|+1$ interesting components in $\mathcal{C}_i$. This still costs at most $(|Y|+1)-1 = |Y|$ vertex deletions, as claimed.

Let $Y'$ contain all the endpoints of $S$-edges that we delete to get an $S$-forest. We know that $|Y'| \leq |Y|$ and thus $|(X \setminus Y) \cup Y'| \leq |X|$. Moreover, by the initial considerations, we know that $X' = (X \setminus Y) \cup Y'$ is a feasible solution as $G - X'$ is an $S$-forest. If $|Y'| < |Y|$, including the case that $Y' = \emptyset$, then $|X'| < |X|$ as $Y \neq \emptyset$. But, this contradicts the optimality of $X$ (required for being a dominant solution). If $|Y'| = |Y|$ then $Y' \neq \emptyset$ and $X'$ is an optimal solution that contains more vertices of $T \supseteq Y'$, contradicting the choice of $X$ as a dominant solution. Thus, every non-empty set $Y$ must see at least $|Y|+2$ interesting connected components. This concludes the proof. ■

Now, we are able to give the proof of Lemma 13.2. The argument relies on Hall's Theorem (Theorem 2.1) and is similar to the one for Deletable Terminal Multiway Cut [KW12].

**Proof of Lemma 13.2.** We know that every non-empty set $Y \subseteq X_0$ sees at least $|Y|+2$ interesting components of $G - X - S$ by Lemma 13.3. To prove the existence of the required path packing we construct a bipartite graph where one side consists of the interesting components and the other side consists of the set $X_0$ and two copies $x', x''$ of the vertex $x \in X_0$. We connect $v \in X_0$ with an interesting component $K$ if $v$ sees $K$ and we connect $x'$ and $x''$ with the same interesting components as $x$. For this bipartite graph it holds that for all sets $Y \subseteq X_0 \cup \{x', x''\}$, the size of $N(Y)$ is at least $|Y|$: This holds trivially for $Y = \emptyset$. Thus, assume there exists a non-empty set $Y \subseteq X_0 \cup \{x', x''\}$ such that $|N(Y)| < |Y|$. But then we have $|N(Y \setminus \{x', x''\})| \leq |N(Y)| < |Y| \leq |Y \setminus \{x', x''\}| + 2$, which is a contradiction to Lemma 13.3.

Since Hall's condition is satisfied there exists a matching $M$ that saturates the set $X_0 \cup \{x', x''\}$. This matching gives rise to a path packing from $T$ to $X$ where exactly three paths end in $x$ and no other vertices occur in more than one path: For each vertex $v \in X \cap T$ we pick the path of length zero that consists only of $v$. For each edge $\{K, v\}$ in the matching $M$, where $v \in X_0 \cup \{x', x''\}$, we pick an arbitrary path from a terminal $t \in V(K) \cap (T \setminus X)$ to $v$ that uses only vertices from $V(K) \cup \{v\}$. (For $v \in \{x', x''\}$ let the path end in $x$ and use only vertices in $V(K) \cup \{x\}$.) Because $K$ is an interesting component a terminal $t \in V(K) \cap (T \setminus X)$ must exist, and because $K$ is a component of $G - X - S$ the path contains no other vertices of $X$. Similarly, the path cannot contain $S$-edges between vertices of $K$, and its final edge to $v$ cannot be in $S$ because $v \in X_0 = X \setminus T$, i.e., because $v$ is not an endpoint of any $S$-edge. Moreover, since each interesting component is matched to a single vertex $v \in X_0 \cup \{x', x''\}$, all the paths are vertex-disjoint except for the three paths that share their endpoint $x$.

This path packing, including the trivial paths from $X \cap T$ to $X \cap T$, contains $|X| + 2$ paths from $T$ to $X$ in $G - S$ that are vertex-disjoint except for the three paths sharing endpoint $x$. By construction, there is at most one path to any vertex of $X_0$ starting in any interesting component $K$ of $G - X - S$, because the components are used according to the matching $M$. All further paths are of length zero, consisting of only one vertex in $X \cap T$ and are, thus, not contained in components of $G - X - S$. ■

**Setting up the gammoid.** The gammoid $M$ that we use is the direct sum of two gammoids $M_1$ and $M_2$. To construct gammoid $M_1$ we define a graph $G_1 = (V_1, E_1)$ that is obtained from $G - S$ by adding two so-called *sink-only copies* $v'$ and $v''$ for every vertex $v \in V$. A sink-only copy of a vertex $v$ is a vertex $v'$ (or $v''$) that has a directed edge $(u, v')$ for each edge $\{u, v\}$ in $G - S$; these were already used in previous work [KW12]. Note that adding sink-only copies of vertices does not affect the possible path packings to other vertices since they can only be endpoints of paths. However, they are convenient to capture multiple vertex-disjoint paths that, intuitively, end in the same vertex. The matroid $M_1$ is defined as the gammoid on $G_1$ with sources $T = V(S)$ and ground set $V_1 = \{v, v', v'' \mid v \in V\}$. Note that the sink-only copies of vertices in $T$ are not sources of $M_1$. The rank of matroid $M_1$ is $|T|$, because the set of all trivial paths is independent and at most $|T|$ vertices can be linked to $T$.

Matroid $M_2$ is the gammoid on the directed graph $G_2 = K_{k,n} = (S_2 \dot\cup \hat{V}, E_2)$ with sources $S_2$ and ground set $\hat{V} = \{\hat{v} \mid v \in V\}$. The edges in $E_2$ are directed from $S_2$ to $\hat{V}$. The rank of $M_2$ is $k = |S_2|$ because no more than $|S_2|$ vertices can be linked to $S_2$ and every set of at most $k$ vertices of $\hat{V}$ is linked to $S_2$. In other words, gammoid $M_2$ is a uniform matroid of rank $k$ and a (deterministic) matrix representation can be obtained by using a Vandermonde matrix.

For the application of Lemma 12.10 we will use the matroid $M = M_1 \oplus M_2$, which has rank $|T| + k$. (Matroid $M$ can also be seen as a gammoid on the graph $G_1 \dot\cup G_2$ with appropriate sources and ground set but we prefer the explicit direct sum and the implied block-diagonal representation obtained below.) Representations $A_1$ and $A_2$ for both $M_1$ and $M_2$ can be computed by a randomized polynomial-time algorithm with exponentially small error chance [Mar09]; hence we get a representation for $M$ by $\operatorname{diag}(A_1, A_2)$, i.e., the block-diagonal matrix with blocks $A_1$ and $A_2$. We may assume that $A_1$ has $|T|$ rows and $A_2$ has $k$ rows since this could be achieved by Gaussian elimination (cf. [Mar09]).

**Applying the representative set lemma.** Let $\mathcal{T} := \{\{v', v'', \hat{v}\} \mid v \in V\}$. For clarity, by the above notation, this means that $v', v'' \in V_1$ and $\hat{v} \in \hat{V}$ for each $v \in V$. Using Lemma 12.10 we will prove that we can compute in randomized polynomial time a $(|T|+k-3)$-representative subset $\mathcal{T}'$ of $\mathcal{T}$ that contains for all $x \in X_0 = X \setminus T$ the set $\{x', x'', \hat{x}\}$, where $X$ is any dominant solution for $(G, S, k)$. Lemma 12.10 guarantees that $|\mathcal{T}'| \in \mathcal{O}((|T| + k)^3) = \mathcal{O}(|S|^3)$, since we can compute a matrix representation of $M$ in randomized polynomial time as described above. We will see later that we can find a $(|T| + k - 3)$-representative set of size $\mathcal{O}(|S|^2 k)$ by a careful look at the proof of Lemma 12.10, using the fact that the matroid $M$ is the direct sum of two gammoids and that all sets $\{v', v'', \hat{v}\}$ in $\mathcal{T}$ have two elements from the first and one element from the second gammoid. A similar argument for getting a smaller representative set was already used by Kratsch and Wahlström [KW12].

To ensure that all sets $\{x', x'', \hat{x}\}$ with $x \in X_0$ are in $\mathcal{T}'$ we have to show that for each such set $\{x', x'', \hat{x}\}$ there exists an independent set $I$ of size at most $|T| + k - 3$ in $M$ such that $\{x', x'', \hat{x}\}$ uniquely extends $I$ among triplets in $\mathcal{T}$. This directly implies that $\{x', x'', \hat{x}\}$ must be in every $(|T| + k - 3)$-representative set $\mathcal{T}'$ of $\mathcal{T}$.

**Lemma 13.4.** *Let $X$ be a dominant solution for $(G, S, k)$ and let $T = V(S)$. For all $x \in X_0 = X \setminus T$ there exists an independent set $I$ of size at most $|T| + k - 3$ in $M$ such that $\{x', x'', \hat{x}\}$ uniquely extends $I$.*

**Proof.** Let $x$ be an arbitrary vertex of $X_0$. In a first step we define an independent set $I$ and show in a second step that $\{x', x'', \hat{x}\}$ uniquely extends $I$. Applying Lemma 13.2 implies the existence of a path packing $\mathcal{P}$ of $|X| + 2$ paths from $T$ to $X$ in $G - S$ that are vertex-disjoint except for three paths ending in $x$ and such that each connected component of $G - X - S$ is intersected by at most one path of $\mathcal{P}$. This directly implies that there exists a path packing $\mathcal{P}_1$ in $G_1$ from $T$ to $X \cup \{x', x''\}$ that is (fully) vertex-disjoint. We retain the property that at most one path intersects the vertex set of any component of $G - X - S$, but note that we do not get exactly the same property for $G_1 - X$ because of the still present sink-only copies of vertices in $X$. The latter point will be no problem and should mainly explain why we need to talk about $G - X - S$ and not only $G_1$. Note that $G - S$ and $G_1$ by construction share the vertex set $V$ to be able to refer to connected components of $G - X - S$ and the graph $G_1$ underlying the gammoid $M_1$.

While we do not know the paths in $\mathcal{P}_1$ entirely, we know for sure that no vertex of $X \cup \{x', x''\}$ can be an internal vertex of any path in $\mathcal{P}_1$ because there is a path ending in each of those vertices. Similarly, we may assume that no vertex of $T$ is an internal vertex of any path of $\mathcal{P}_1$: If not then any path $P \in \mathcal{P}_1$ with an internal vertex in $T$ can be shortened to start in that vertex. This argument cannot be repeated indefinitely as the paths get shorter each time. There is still at most one path intersecting the vertex set of any component of $G - X - S$.

Now, define $T' \subseteq T$ as those vertices of $T$ in which no path of $\mathcal{P}_1$ starts. There must be exactly $|T| - |\mathcal{P}| = |T| - (|X| + 2)$ of them since no vertex of $T$ is internal. Moreover, for each component $K$ of $G - X - S$, the set $T'$ contains all but at most one vertex of $T \cap V(K)$: At most one path of $\mathcal{P}_1$ can start in $T \cap V(K)$ and no vertex can be internal. This will be important for proving the claim below.

Clearly, the set $T' \cup X \cup \{x', x''\}$ is independent in $M_1$ because an appropriate path packing $\mathcal{P}'$ can be obtained from $\mathcal{P}_1$ by adding length zero paths for each vertex $v \in T'$. The set $\hat{X} = \{\hat{x} \mid x \in X\} \subseteq \hat{V}$ is clearly independent in $M_2$ since it has size at most $k$. Thus, the set $I' = T' \cup X \cup \{x', x''\} \cup \hat{X}$ is independent in $M = M_1 \oplus M_2$. Define $I$ as $I' \setminus \{x', x'', \hat{x}\}$, i.e., $I = T' \cup X \cup (\hat{X} \setminus \{\hat{x}\})$. The size of $I$ is at most

$$|T'| + |X| + (|\hat{X}| - 1) = |T| - (|X| + 2) + |X| + |X| - 1 = |T| + |X| - 3 \leq |T| + k - 3.$$

Clearly, $\{x', x'', \hat{x}\}$ extends $I$, as $I' = \{x', x'', \hat{x}\} \cup I$ is independent and both are disjoint by choice of $I$. We show that no other set $\{v', v'', \hat{v}\} \in \mathcal{T}$ extends $I$.

Claim 13.5. *If $\{v', v'', \hat{v}\} \in \mathcal{T}$ extends $I$ then $v = x$.*

Proof. Suppose that the set $\{v', v'', \hat{v}\} \in \mathcal{T}$ extends $I$. Clearly, this implies that $v \notin X \setminus \{x\}$ because otherwise $\{v', v'', \hat{v}\}$ would not be disjoint from $\hat{X} \setminus \{\hat{x}\} \subseteq I$. Thus, $v \in V \setminus (X \setminus \{x\})$.

Assume, for contradiction, that $v \in V \setminus X$, i.e., that $v \neq x$. We know that $\{v', v'', \hat{v}\} \cup I$ is independent in $M$, so $I_1 := I \cap V_1$ must be independent in $M_1$. Thus, there exists a collection $\mathcal{P}''$ of $|I_1|$ vertex-disjoint paths from $T$ to $I_1$ in $G_1$. Because $X \subseteq I_1$, the paths, say $P_{v'}$ and $P_{v''}$, from $T$ to $\{v', v''\}$ cannot have internal vertices from the set $X$. Furthermore, they cannot have other sink-only copies as internal vertices. Since $v \in V \setminus X$, this implies that $P_{v'}$ and $P_{v''}$ are entirely contained in some component $K_1$ of $G_1 - (X \cup \{x', x'' \mid x \in X\})$. Recall, component $K_1$ corresponds to a connected component $K$ of $G - X - S$ but also has sink-only copies of each vertex. Now, by the choice of $T'$, we have that all except at most one vertex of $T \cap V(K)$ for each connected component of $G - X - S$ is contained in $T'$. This is also true for $T \cap V(K_1)$

as $V(K_1) \cap V = V(K)$. Thus, in $\mathcal{P}''$ there is a path $w$ of length zero for each vertex $w$ in $T' \cap V(K_1)$, leaving at most one vertex of $T$ to start paths to $\{v', v''\}$. This is a contradiction because $P_{v'}$ and $P_{v''}$ are entirely contained in $K_1$ and fully vertex-disjoint.

Thus, if $v \in V \setminus X$ then $\{v', v''\} \cup I_1$ is not independent in $M_1$ and, hence, $\{v', v'', \hat{v}\}$ does not extend $I$ in $M$. Together with the first paragraph this implies that $v = x$, as claimed. □

Overall, we showed that the set $\{x', x'', \hat{X}\} \in \mathcal{T}$ uniquely extends the independent set $I$ that we have constructed using Lemma 13.2. This completes the proof. ■

Now, we know that for every vertex $x \in V \setminus T$ that is a vertex in a dominant solution the set $\{x', x'', \hat{x}\}$ is contained in every $(|T| + k - 3)$-representative set $\mathcal{T}'$ for $\mathcal{T}$. If we define $V(\mathcal{T}') = \{v \in V \mid \{v', v'', \hat{v}\} \in \mathcal{T}'\}$ then this implies that $X_0 \subseteq V(\mathcal{T}')$ for each dominant solution $X$. Thus, every dominant solution $X$ is contained in $V(\mathcal{T}') \cup T$.

**Shrinking the input graph to $|V(\mathcal{T}') \cup T|$ vertices.** So far, we have shown that if there exists a solution for $(G, S, k)$, then there exists a solution that is completely contained in $W := V(\mathcal{T}') \cup T$ because every dominant solution is contained in $W$. Using this, we can make all vertices in $V \setminus W$ undeletable. We achieve this by applying a variant of the torso operation to the vertex set $W$ in $G$. We construct a graph $G'$ as follows: The graph $G'$ has vertex set $W$ and is derived from $G[W]$ by adding an edge between two different vertices $u$, $v \in W$, if there exists a $u$,$v$-path in $G$ with internal vertices from $V \setminus W$ and, if $\{u, v\}$ is either a non-edge in $G[W]$ or an $S$-edge. To clarify, this allows double edges in $G'$ when one edge is contained in $S$, but we never create loops. Furthermore, all edges of $S$ are preserved in $G'$ because $T \subseteq W$.

**Lemma 13.6.** *The instance $(G', S, k)$ of* Edge Subset FVS *has a solution if and only if the instance $(G, S, k)$ of* Edge Subset FVS *has a solution.*

It follows from Lemma 13.6 that $(G', S, k)$ is an equivalent instance and the graph of this instance contains at most $|W|$ vertices. This completes the kernelization. The correctness of Lemma 13.6 follows from the fact that the torso operation preserves the separators that are contained in $W$ (cf. [MOR13]). For completeness we give a short proof of the lemma.

**Proof of Lemma 13.6.** Let $X$ be a solution for $(G', S, k)$. We prove that $X$ is also a solution for $(G, S, k)$ by contradiction. Assume that $X$ is not a solution for $(G, S, k)$. Then there exists an $S$-cycle $C = v_1 e_1 v_2 e_2 \dots v_l e_l v_1$ in $G - X$. Note that $S \subseteq E(G')$, because $T = V(S) \subseteq W$ and therefore at least two vertices of $C$ are contained in $W$. Now we modify $C$ to obtain an $S$-cycle $C'$ in $G'$. Let $v_i, v_j \in W \cap C$ two vertices of the cycle with $1 \le i < j \le l$ such that $\{v_{i+1}, \dots, v_{j-1}\} \subseteq V \setminus W$. By definition, there exists an edge $\{v_i, v_j\}$ in $G'$ and using these edges we obtain a cycle $C'$. Note that $C'$ contains no vertex of $X \subseteq W$ and contains the same edges from $S$ that $C$ contains. Thus, $C'$ is an $S$-cycle in $G' - X$ which contradicts the assumption that $X$ is a solution for $(G', S, k)$.

For the other direction we assume that $(G, S, k)$ has a solution. Then there also exists a dominant solution $X$ for $(G, S, k)$ and we know that $X \subseteq W$. Again we prove that $X$ is also a solution for $(G', S, k)$ by contradiction. Assume that $X$ is not a solution for $(G', S, k)$. Then there exists a path $P'$ between the endpoints of an edge $e = \{x, y\} \in S$ in $G' - X$ that does not use the edge $e$, i.e., $P'$ is an $x, y$-path in $G - X'$ that does not contain edge $e$. We modify $P'$ to obtain a path $P$ in $G$ that does not contain the edge $e$. If $P'$ uses an edge $\{u, v\}$ that is not contained in $G$, then there exists a $u, v$-path in $G$ with internal vertices from $V \setminus W$. Crucially, $V \setminus W$ is disjoint from $X$ so this replacement still yields a walk from $x$ to $y$ that avoids $X$. Overall, we obtain a walk from vertex $x$ to vertex $y$ in $G$ that does not contain $e$ as an edge and that avoids $X$. This walk contains a path $P$ from $x$ to $y$ and this path together with the edge $e$ is an $S$-cycle in $G - X$ which is a contradiction to the assumption that $X$ is a solution for the instance $(G, S, k)$. ∎

So far we have a kernelization that creates an equivalent instance $(G', S, k)$ such that $G'$ has $|W|$ vertices. As mentioned above, Lemma 12.10 guarantees that $|W| \in \mathcal{O}(|S|^3)$ and this implies a polynomial kernel for Edge Subset FVS parameterized by $|S|$. If we use the fact that the gammoid $M$ is the direct sum of two gammoids $M_1$ and $M_2$, and that all sets $\{v', v'', \hat{v}\} \in \mathcal{T}$ contain exactly two elements of $M_1$ and one element of $M_2$, then we can prove that $|W| \in \mathcal{O}(|S|^2 k)$, which is an improvement for all nontrivial instances with parameter $k < |S|$.

**Lemma 13.7.** *Let $M = M_1 \oplus M_2$ be the gammoid of rank $|T| + k$ as defined above and $\mathcal{T} = \{I_1, I_2, \ldots, I_t\}$ be a subset of independent sets of $M$ that we used for the kernelization. Let $A$ be represented by $\operatorname{diag}(A_1, A_2)$ as above. If $|\mathcal{T}| > \binom{|T|}{2} \cdot \binom{k}{1}$, then there exists a set $I \in \mathcal{T}$ such that $\mathcal{T} \setminus \{I\}$ is $(|T| + k - 3)$-representative for $\mathcal{T}$.*

The proof of Lemma 13.7 is similar to Marx [Mar09, Lemma 4.2]. We additionally use the fact that $M$ is the direct sum of two matroids (gammoids) to obtain that the vectors in the exterior algebra, which represent the sets in $\mathcal{T}$, span a space of smaller dimension.

**Proof of Lemma 13.7.** Let $U$ be the ground set of the matroid $M$ which equals the set of columns of $A$. For each element $e \in U$, let $x_e$ be the corresponding $(|T| + k)$-dimensional column vector of $A$ and for each $i \in [t]$ let $w_i = \bigwedge_{e \in I_i} x_e$ be a vector in the exterior algebra of the linear space $F^{|T|+k}$. Every $w_i$ is the wedge product of three vectors where exactly two are column vectors from $\left(\begin{smallmatrix} A_1 \\ 0 \end{smallmatrix}\right)$ and one is a column vector from $\left(\begin{smallmatrix} 0 \\ A_2 \end{smallmatrix}\right)$. The two vectors corresponding to $\left(\begin{smallmatrix} A_1 \\ 0 \end{smallmatrix}\right)$ can only span a space of dimension $\binom{|T|}{2}$ and the vector corresponding to $\left(\begin{smallmatrix} 0 \\ A_2 \end{smallmatrix}\right)$ can only span a space of dimension $\binom{k}{1}$. Thus, the $w_i$'s, with $i \in [t]$, span a space of dimension at most $\binom{|T|}{2} \cdot \binom{k}{1}$. If $|\mathcal{T}| > \binom{|T|}{2} \cdot \binom{k}{1}$, then the $w_i$'s are not independent and there exists some vector $w_l$, with $l \in [t]$, that can be expressed as a linear combination of the other vectors.

One can show, analogously to Marx [Mar09, Lemma 4.2], that $\mathcal{T} \setminus \{I_l\}$ is $(|T| + k - 3)$-representative for $\mathcal{T}$. We replicate this proof for convenience of the reader. Assume that there exists an independent set $Y$ of size at most $|T| + k - 3$ in $M$ such that $I_l$ extends $Y$ and no other set $I_i$, with $i \in [t]$ and $i \neq l$, extends $Y$. Let $y = \bigwedge_{e \in Y} x_e$. One property of the wedge product is that the product of some vectors in $F^{|T|+k}$ is zero if and only if they are not independent. Therefore it holds that $w_l \wedge y \neq 0$ and $w_i \wedge y = 0$ for every $i \neq l$. But $w_l$ is a linear combination of other $w_i$'s and by the multi-linearity of the wedge product we get that $w_l \wedge y \neq 0$ is a linear combination of the values $w_i \wedge y = 0$ for $i \neq l$, which is a contradiction. This shows that $\mathcal{T} \setminus \{I_l\}$ is $(|T| + k - 3)$-representative for $\mathcal{T}$ and concludes the proof. ∎

As mentioned above, Marx [Mar09] showed that one can find in randomized polynomial time a matrix with $r(M)$ rows that represents a given gammoid $M$. We can make this proof algorithmic in the same way Marx did [Mar09, Lemma 4.2]. Combined with Lemma 13.7 it follows directly that we can find a $(|T| + k - 3)$-representative subset $\mathcal{T}'$ for $|\mathcal{T}|$ whose size is at most $\binom{|T|}{2} \cdot \binom{k}{1} \in \mathcal{O}(|S|^2 k)$. This leads to a polynomial kernel with $\mathcal{O}(|S|^2 k)$ vertices for Edge Subset FVS parameterized by $|S|$ and $k$.

## 13.4. Reducing the Size of $S$

We have seen that Edge Subset FVS parameterized by $|S|$ and $k$ has a polynomial kernel. Now, the goal is to reduce the size of the set $S$ until $|S|$ is polynomially bounded in $k$. This will lead to a polynomial kernel of Edge Subset FVS parameterized by $k$.

To begin, we do some initial modifications to ensure that we can always find a solution of size at most $k$ that contains no vertex of the set $V(S)$, if one exists. For this we first delete all vertices $v \in V$ with the property that $e = \{v, v\} \in S$ is a loop in $G$. Since the vertex $v$ must be in any solution, we decrease the value $k$ by one. Next we delete all remaining loops, because these loops are not in $S$ and cannot be contained in any $S$-cycle. We also reduce the number of edges between two vertices $v, w \in V(G)$. If no edge that is incident with $v$ and $w$ is contained in the set $S$, then we delete all except one edge. On the other hand, if at least one edge between $v$ and $w$ is contained in $S$, then we delete all except two edges, so that one of these edges is contained in $S$ and the other not. In the next step we add for every edge $e = \{v, w\} \in S$ two new vertices $v_e, u_e$ to the graph, subdivide the edge $e$ into three edges $\{v, v_e\}, \{v_e, w_e\}, \{w_e, w\}$, and edit $S$ by replacing edge $e$ by the edge $\{v_e, w_e\}$ in $S$. If a solution $X$ of Edge Subset FVS contains a vertex $x_e \in V(S)$, then we can instead add the vertex $x$ to $X$ and delete $x_e$ from $X$, because every cycle that contains vertex $x_e$ also contains vertex $x$. Hence, we can always find an optimal solution that is disjoint from $V(S)$. After these modifications, we obtain a graph where every endpoint of an $S$-edge has degree at most two and every vertex is the endpoint of at most one $S$-edge. Note that these two properties are sufficient to ensure that we can always find a solution that is disjoint from $V(S)$. All our reduction rules, except Reduction Rule 13.1, will preserve these two properties, because we never add $S$-edges to the graph or increase the degree of a vertex; we only delete edges from the set $S$ (not necessarily $G$) or vertices from $G$. Note that it is no problem that Reduction Rule 13.1 does not preserve this property, because this reduction rule returns a trivially false instance and completes the kernelization.

Let $(G, S, k)$ be an instance of Edge Subset FVS, such that $G$ is a graph with the above properties. Analogous to the paper of Cygan et al. [CPPW13b] we consider a solution $Z$ of the Edge Subset FVS instance $(G, S, k)$, with the difference that our solution is an 8-approximation of the problem, to reduce the size of $S$. Even et al. [ENZ00] showed that there exists an 8-approximation algorithm for Subset FVS. Since there are polynomial-time target-value preserving reductions between Subset FVS and Edge Subset FVS (cf. [CPPW13b]), we can compute in polynomial time an 8-approximation for Edge Subset FVS and we can assume that $Z \cap V(S) = \emptyset$. If $|Z| > 8k$, then we can stop immediately because no solution of size at most $k$ can exist. On the other hand, if $|Z| \leq k$, then $Z$ is a solution and we are done.

The set $Z$ is a feasible solution to Edge Subset FVS on $(G, S, |Z|)$. This implies that every edge $e \in S$ is a bridge in $G - Z$. In the next step we also remove all edges in $S$ from $G - Z$. Every connected component in $G - Z - S$ contains no edge from $S$ and, following Cygan et al. [CPPW13b], we call such a component a *bubble*. We denote the set of bubbles by $\mathcal{D}_Z$ and define a graph $H_Z = (\mathcal{D}_Z, E_{\mathcal{D}_Z})$ whose vertices are bubbles and with bubbles $I$ and $J$ being adjacent, i.e., $\{I, J\} \in E_{\mathcal{D}_Z}$, if and only if the components $I$ and $J$ are connected by an edge from $S$. The graph $H_Z$ is a forest, because $Z$ is a solution for $(G, S, |Z|)$ and a cycle in $H_Z$ would give rise to an $S$-cycle in $G - Z$. Similarly, no two bubbles can be connected by more than one edge of $S$. By $V_I$ we denote the vertices that are contained in the bubble $I$. Since $|E(V_I, V_J) \cap S| \leq 1$ for all $I, J \in \mathcal{D}_Z$ and equality holds if and only if $\{I, J\} \in E_{\mathcal{D}_Z}$, we can associate an edge $e = \{I, J\} \in E_{\mathcal{D}_Z}$ with the one edge $e_S = \{v_I, v_J\}$ in $E(V_I, V_J) \cap S$. If we add the vertex set $Z$ and all edges $\{z, I\}$ with the property that $z \in Z, I \in \mathcal{D}_Z$ and $E(z, V_I) \neq \emptyset$ to the graph $H_Z$ we obtain a graph $H_Z^+$ that contains $S$-cycles. Note that every $S$-cycle must contain a vertex of the set $Z$. We partition the set of bubbles according to the number of bubbles they are connected with.

**Definition 13.8.** A bubble $I \in \mathcal{D}_Z$ is called

(i) solitary, if $\deg_{H_Z}(I) = 0$;

(ii) leaf, if $\deg_{H_Z}(I) = 1$; and

(iii) inner, if $\deg_{H_Z}(I) \geq 2$.

By $\mathcal{D}_Z^s, \mathcal{D}_Z^l, \mathcal{D}_Z^i$ we denote the corresponding sets of bubbles.

Let $X \subseteq V \setminus V(S)$ be a superset of $Z$. We define the graphs $H_X$, $H_X^+$ as well as the sets $\mathcal{D}_X, E_{\mathcal{D}_X}$ analogously to the graphs $H_Z, H_Z^+$ and the sets $\mathcal{D}_Z, E_{\mathcal{D}_Z}$. Observe that the number of edges in $S$ is equal to the number of edges in the graph $H_Z$, because $Z$ is a feasible solution to $(G, S, |Z|)$ that is disjoint from $V(S)$ and because two bubbles $I$, $J$ in the forest $H_Z$ are only connected when there exists an $S$-edge in $E(V_I, V_J)$. Since

$H_Z$ is a forest, the number of edges in $H_Z$ is bounded by the number of non-isolated vertices. Thus, the number of edges in $S$ is at most $|\mathcal{D}_Z \setminus \mathcal{D}_Z^s|$.

So far our setup is essentially the same as the one used by Cygan et al. [CPPW13b]. However, instead of an 8-approximate solution they use the framework of iterative compression, which provides a solution $Z$ of size $k+1$ and leaves them with the task of reducing the number of $S$-edges for the problem of finding a solution $Z^*$ that is *disjoint* from $Z$. Moreover, it suffices for them to consider the case that every feasible solution (if one exists) is disjoint from $Z$. In this setting they are able to reduce to an equivalent instance (or find that some assumption was violated) with only $\mathcal{O}(k^3)$ edges in $S$.

Thus, while many relevant structures like $z$-flowers or parallel $x, y$-paths containing $S$-edges are the same, many things have to be handled differently. In particular, if we find that at least one out of two vertices $x, y \in Z$ must be in the solution then we cannot stop (like Cygan et al. [CPPW13b]) but need to continue and use this information in a more direct way. Cygan et al. [CPPW13b] can stop whenever they find such a pair $\{x, y\}$, because they solve Disjoint Edge Subset FVS during the iterative compression step and are only interested in so-called maximal instances where *every* solution of the instance is disjoint from $Z$.

During the reduction we detect certain pairs $\{x, y\}$ of different vertices with the property that each solution of size at most $k$ must contain at least one of the vertices (if one exists). We store this fact as a *pair-constraint*. We keep and enforce this information in the final instance, unless we decide earlier to delete $x$ or $y$. By $\mathcal{P}$ we denote the set of pair-constraints that we have found so far. We can interpret this set as a set of edges and by $V(\mathcal{P})$ we denote all vertices that are contained in a pair-constraint. Note that vertices from the set $V(S)$ are never contained in a pair-constraint from $\mathcal{P}$, because there always exists a solution that is disjoint from $V(S)$. We need the set $\mathcal{P}$ to detect edges in $S$ that may be safely deleted. To this end, we generalize the Edge Subset FVS problem by adding a set of pair-constraints $\mathcal{P}$ to the input. We call this problem Pair-Constrained Edge Subset FVS.

Pair-Constrained Edge Subset Feedback Vertex Set
**Parameter:** $k$
**Input:** An undirected graph $G$, a set $S \subseteq E$ of edges, a set $\mathcal{P}$ of pair-constraints with $V(\mathcal{P}) \cap V(S) = \emptyset$ and an integer $k$.
**Question:** Does there exist a set $X \subseteq V$ of size at most $k$ such that $G - X$ contains no $S$-cycle and such that for each pair-constraint $\{x, y\} \in \mathcal{P}$ we have $x \in X$ or $y \in X$?

Clearly, the instance $(G, S, k)$ of Edge Subset FVS and the instance $(G, S, \emptyset, k)$ of Pair-Constrained Edge Subset FVS are equivalent. Our goal is to reduce the size of $S$ by detecting $S$-edges that we can delete from $S$ without changing the outcome. This leads to the following definition:

**Definition 13.9.** Let $(G, S, \mathcal{P}, k)$ be an instance of Pair-Constrained Edge Subset FVS. We call an edge $e \in S$ *irrelevant*, if $X \subseteq V(G)$ is a solution for $(G, S, \mathcal{P}, k)$ if and only if $X$ is a solution for $(G, S \setminus \{e\}, \mathcal{P}, k)$.

Note that if two different $S$-edges $e$ and $e'$ are irrelevant in $(G, S, \mathcal{P}, k)$, then $e'$ is not necessarily irrelevant in $(G, S \setminus \{e\}, \mathcal{P}, k)$. In addition we do not expect to find all irrelevant edges or pair-constraints.

**The reduction rules.** We now present our reduction rules. Throughout we assume that always the lowest numbered applicable reduction rule is applied first. Correctness and efficiency of the overall reduction process will be proved later.

Let $(G, S, \mathcal{P} = \emptyset, k)$ be an instance for Pair-Constrained Edge Subset FVS and let $Z$ be an 8-approximation of this problem with $k < |Z| \leq 8k$ that is disjoint from $V(S)$. In the following the graphs $G - Z$, $G - Z - S$, $H_Z$, and $H_Z^+$ are always defined with respect to the current instance $(G, S, \mathcal{P}, k)$ of Pair-Constrained Edge Subset FVS. Note that $Z \subseteq V$ and we delete a vertex from $Z$ if we delete the corresponding vertex in $V$. Furthermore, we delete a pair-constraint $\{x, y\}$ from $\mathcal{P}$ if we delete $x$ or $y$ from $G$.

**Reduction Rule 13.1.** If $k < 0$, or if $k = 0$ and there exists an $S$-cycle, then we reduce $(G, S, \mathcal{P}, k)$ to some trivial false instance, i.e., $G' := (\{x\}, \{e = \{x, x\}\})$, $S' := \{e\}$, $\mathcal{P}' := \emptyset$ and $k' := 0$.

**Reduction Rule 13.2.** Delete all bridges and all connected components in $G$ not containing any edge from $S$.

**Reduction Rule 13.3.** If there exists an edge $e \in S$ such that $e$ is a bridge in $(V, E \setminus (S \setminus \{e\}))$, then we reduce to $S' = S \setminus \{e\}$.

Note that, if Reduction Rule 13.3 deletes an edge $e$ from $S$ then this edge is still contained in $G$. The Reduction Rules 13.2 and 13.3 ensure that each bubble $I \in \mathcal{D}_Z$ is adjacent to a vertex in $Z$ in the graph $H_Z^+$, i.e., for all bubbles $I \in \mathcal{D}_Z$ we have that $E_{H_Z^+}(I, Z) \neq \emptyset$: Since Reduction Rule 13.2 is not applicable every bubble $I \in \mathcal{D}_Z$ must be adjacent to a bubble $J \in \mathcal{D}_Z \setminus I$, or a vertex in $Z$; otherwise $G[V_I]$ would be a connected component of $G$ that does not contain any edge from $S$ ($V_I$ was deleted in Reduction Rule 13.2). From Reduction Rule 13.3 it follows that a bubble $I \in \mathcal{D}_Z$ must be adjacent to a vertex in $Z$; otherwise every edge $e \in E_G(V_I, N(V_I)) \cap S$ would be a bridge in $(V, E \setminus (S \setminus \{e\}))$.

**Reduction Rule 13.4.** If there exists a vertex $v$ in the set $V(\mathcal{P})$ that is contained in at least $k + 1$ pair-constraints of $\mathcal{P}$, then we reduce to $G' = G - v$ and $k' = k - 1$.

**Reduction Rule 13.5.** If $|\mathcal{P}| > k^2$ (and Reduction Rule 13.4 is not applicable), then we reduce $(G, S, \mathcal{P}, k)$ to some trivial false instance.

**Reduction Rule 13.6.** If there exists a $z$-flower of order $k+1$ in $G$ for a vertex $z \in Z$, then we reduce to $G' := G - z$ and $k' := k - 1$.

For the next reduction rules we need a maximal matching $M$ in $H_Z$ that saturates all inner bubbles $\mathcal{D}_Z^i$ in $H_Z$. We will show later (Lemma 13.21) that such a matching exists. Note that two adjacent leaf bubbles $I_1, I_2$ are not adjacent to an inner bubble and form a $K_2$ in $H_Z$, hence the edge $\{I_1, I_2\} \in E_{\mathcal{D}_Z}$ is contained in every maximal matching in $H_Z$. We use this matching to detect pair-constraints in $Z$. To this end we introduce the following definition: Let $e = \{I, J\}$ be an edge in the matching $M$. We say that edge $e$ *sees* the pair $\{x, y\}$ of different vertices $x, y \in Z$, if $\{I, x\}, \{J, y\} \in E(H_Z^+)$ or $\{I, y\}, \{J, x\} \in E(H_Z^+)$. Similar, we say that edge $e$ *sees* the vertex $x \in Z$, if $\{I, x\}, \{J, x\} \in E(H_Z^+)$.

**Reduction Rule 13.7.** If at least $(k + 2)$ edges in $M$ see a pair $\{x, y\}$ of different vertices in $Z$, then we add $\{x, y\}$ to the set of pair-constraints $\mathcal{P}$.

**Reduction Rule 13.8.** If there exists an edge $e \in M$ such that $e$ sees no single vertex $z \in Z$ and for every pair $\{x, y\}$ seen by $e$ the pair $\{x, y\}$ is a pair-constraint in $\mathcal{P}$, then we remove $e_S$ from $S$ and $e$ from $M$. (Recall: If $e = \{I, J\} \in M \subseteq E(H_Z)$, then $e_S$ is the unique edge in $E(V_I, V_J) \cap S$.)

The matching $M$ is always recomputed if, through application of reduction rules, it does no longer saturate every inner bubble or is not maximal when testing whether Reduction Rules 13.7 or 13.8 apply (i.e., if the preceding reduction rules do not apply). If $M$ does saturate all inner bubbles but neither Reduction Rule 13.7 nor 13.8 applies then, as we will prove later, this implies $|M| \in \mathcal{O}(k^3)$ and, hence, that there are at most $2|M| \in \mathcal{O}(k^3)$ inner bubbles.

Let $L = \mathcal{D}_Z^l \setminus V(M)$ be the set of leaf bubbles that are exposed by $M$. Because the matching $M$ saturates at least all inner bubbles of $H_Z$, we know that the number of non-isolated vertices in the forest $H_Z$ is at most $2|M| + |L|$. Since the number of edges in a forest is bounded by the number of non-isolated vertices and $|S| = |E_{H_Z}|$, we get $|S| \leq 2|M| + |L|$. Therefore we have to find a reduction rule that reduces the number of leaf bubbles in $L$. Every leaf bubble in $L$ is adjacent to an inner bubble in $H_Z$, because matching $M$ saturates all leaf bubbles that are not adjacent to an inner bubble. To bound the number of leaf bubbles in $L$ we define for each vertex $z \in Z$ a graph $G_z$ with the help of the following two sets. The first one, $L_z = N_{H_Z^+}(z) \cap L$, is the set of all exposed leaf bubbles $I$ that are adjacent to $z$ in $H_Z^+$. The other $V_z^i = \{v \in V \mid \exists J \in N_{H_Z}(L_z) \colon v \in V_J\}$ consists of all vertices that are contained in an inner bubble that is adjacent to a leaf bubble in $L_z$. The graph $G_z$ is defined as follows (see Figure 13.3):

$$\begin{aligned} V(G_z) &= \{z\} \cup L_z \cup V_z^i \\ E(G_z) &= E_{H_Z^+}(z, L_z) \cup (E(G[V_z^i]) \setminus S) \\ &\qquad \cup \{\{I, w\} \mid I \in L_z, w \in V_z^i \text{ and } \exists v \in V_I \colon \{v, w\} \in S\} \end{aligned}$$

In the graph $G_z$ each leaf bubble $I \in L_z$ is a single vertex. We are not interested in the internal structure of leaf bubbles in $L_z$, whereas we are interested in the structure of the inner bubbles that are adjacent to the leaf bubbles in $L_z$. Thus we add the connected

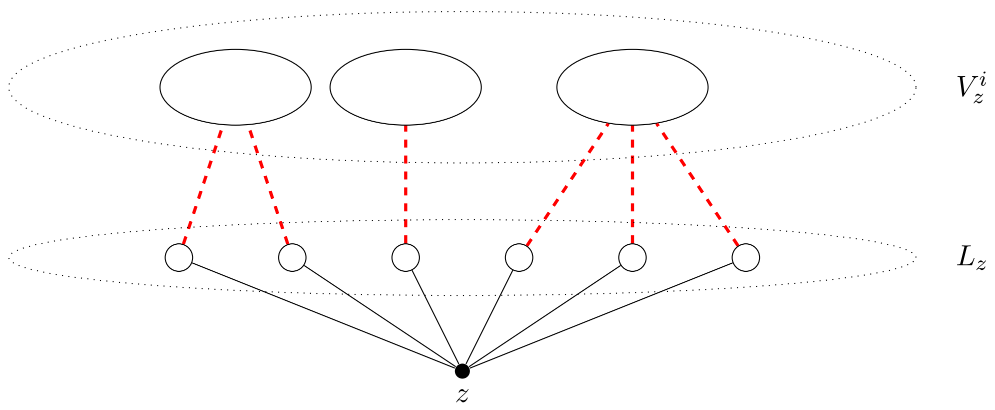



Figure 13.3.: The graph $G_z$, where every ellipse represents a set $V_J$ with $J \in N_{H_Z}(L_z)$. $S$-edges in $G_z$.

component that corresponds to an inner bubble which is adjacent to a bubble in $L_z$ to $G_z$. In order to apply the concept of flowers and $z$-blockers in $G_z$, we need to define which edges of $G_z$ are $S$-edges: An edge $e \in E(G_z)$ is an $S$-edge if one endpoint of the edge $e$ is contained in $L_z$ and the other is contained in $V_z^i$. Recall that $e$ is an edge in $G_z$, because there exists an $S$-edge $e' = \{v, w\}$ in $G$ with $v \in V_I$ and $w \in V_z^i$.

**Lemma 13.10.** *If there exists no $z$-flower of order $k+1$ in $G_z$ for a vertex $z \in Z$, then we can find a $z$-blocker $B_z \subseteq V_z^i \setminus V(S)$ of size at most $2k$ in $G_z$.*

The lemma follows from Theorem 13.1, the preprocessing that every vertex in $V(S)$ has degree two as well as the construction of graph $G_z$.

**Proof of Lemma 13.10.** The maximum number of vertex-disjoint $L_z$-paths in $G_z - z$ is at most $k$. Otherwise, the $L_z$-paths together with vertex $z$ would correspond to a $z$-flower of order $k+1$ in $G_z$ which contradicts the assumption. From Theorem 13.1 it follows that there exists a set $B_z \subseteq V(G_z - z) = L_z \cup V_z^i$ of size at most $2k$ intersecting every $L_z$-path. Since every $S$-cycle through $z$ in $G_z$ must contain an $L_z$-path, the set $B_z$ is a $z$-blocker of size at most $2k$ in $G_z$.

It remains to show that there exists a $z$-blocker $B_z \subseteq V_z^i \setminus V(S)$. First we assume that there exists a vertex $I \in B_z \cap L_z$. From the construction of $G_z$ it follows that every leaf bubble has degree one in $G_z - z$. Thus instead of $I$ we can choose the unique vertex in $N_{G_z}(I) \cap V_z^i$ for the $z$-blocker $B_z$ to obtain that $B_z \subseteq V_z^i$.

In the next step we take care that $B_z$ is also disjoint from $V(S)$. Assume that $B_z$ contains a vertex $v_e \in V(S) \cap V_z^i$. From the preprocessing it follows that we can add $v \in V_z^i \setminus V(S)$ to $B_z$ and delete $v_e$ from $B_z$, because every cycle that contains $v_e$ also contains $v$.

Overall, we showed that there exists a $z$-blocker $B_z \subseteq V_z^i \setminus V(S)$. Since we delete at least as many vertices from $B_z$ as we add to $B_z$ during our construction, it holds that the set $B_z$ is still of size at most $2k$. This concludes the proof. ■

Since no previous reduction rule is applicable and a $z$-flower of order $k+1$ in $G_z$ gives rise to a $z$-flower of order $k+1$ in $G$, we find a $z$-blocker of size at most $2k$ for every vertex $z \in Z$. Let $B = \bigcup_{z \in Z} B_z$ be the union of all $z$-blockers $B_z$ of size at most $2k$. Note that the set $L$ is the union of all sets $L_z$ with $z \in Z$, because every leaf bubble is adjacent to a vertex in $Z$ due to Reduction Rule 13.2, hence $L = \bigcup_{z \in Z} L_z$.

The following lemma provides nice properties of the graph $H_{Z \cup B} = (\mathcal{D}_{Z \cup B}, E_{\mathcal{D}_{Z \cup B}})$ which helps us to bound the number of leaf bubbles in $L \subseteq \mathcal{D}^l_Z$. Recall that the set $\mathcal{D}_{Z \cup B}$ is the set of bubbles in $G - (Z \cup B) - S$ and two bubbles $I, J$ are adjacent in $H_{Z \cup B}$ if and only if $E(V_I, V_J) \cap S \neq \emptyset$.

**Lemma 13.11.** *The graph $H_{Z \cup B}$ has the following properties:*

*(i) For each bubble $I \in \mathcal{D}_{Z \cup B}$ there exists a bubble $J \in \mathcal{D}_Z$, such that $V_I \subseteq V_J$.*

*(ii) For each leaf bubble $J \in \mathcal{D}_Z$ there exists a leaf bubble $I \in \mathcal{D}_{Z \cup B}$, such that $V_I = V_J$.*

*(iii) Let $I, J \in L$ and $K \in \mathcal{D}^i_{Z \cup B}$, such that $\{I, K\}, \{J, K\} \in E_{\mathcal{D}_{Z \cup B}}$. Then for all $z \in Z$ it holds that $z \notin N_G(V_I)$ or $z \notin N_G(V_J)$.*

**Proof.** Property (i) holds because the set $B$ only splits bubbles of $G - Z - S$. We are now looking at deleting $Z \cup B$ from $G - S$ instead of deleting only $Z$ and thus, we do not merge any two bubbles. Property (ii) follows from the fact that the set $B$ is disjoint from the set of leaf bubbles.

Next we show property (iii) by contradiction. We assume that some vertex $z \in Z$ is contained in $N_G(V_I)$ and in $N_G(V_J)$. Then $I$ and $J$ are both vertices of the graph $G_z$ and hence both are contained in the set $L_z$. Consequently, there exists an $L_z$-path from bubble $I$ through bubble $K$ to bubble $J$ in $H_{Z \cup B}$ which can be extended to a $L_z$-path in $G_z$ not containing any vertex in $B$. This contradicts the fact that $B_z \subseteq B$ blocks all $L_z$-paths in the graph $G_z$ and concludes the proof. ∎

From Lemma 13.11 it follows that $L \subseteq \mathcal{D}^l_{Z \cup B}$. Thus, we can use graph $H_{Z \cup B}$ to bound the number of leaf bubbles in $L$. Let $\boldsymbol{I} = \{J \in \mathcal{D}^i_{Z \cup B} \mid E(L, J) \neq \emptyset\}$ be the set of inner bubbles in $H_{Z \cup B}$ that are adjacent to a leaf bubble in $L$. Clearly the number of edges between $\boldsymbol{I}$ and $L$ in $H_{Z \cup B}$ equals the number $|L|$. Instead of again using a matching to reduce this number we consider more carefully the properties of these edges. To this end, we define the property of *seeing* a pair in a slightly different way. Let $e = \{I, J\}$ be an edge in $H_{Z \cup B}$ with $I \in \boldsymbol{I}$ and $J \in L$. We say that $e = \{I, J\}$ with $I \in \boldsymbol{I}$ and $J \in L$ sees the pair $\{x, y\}$ of different vertices $x \in Z \cup B$ and $y \in Z$, if $\{I, x\}, \{J, y\} \in E(H^+_{Z \cup B})$. Observe that a bubble in $L$ is never adjacent to a vertex in $B$ in the graph $H_{Z \cup B}$, because $B \subseteq \bigcup_{z \in Z} V^i_z \setminus V(S)$.

**Reduction Rule 13.9.** If at least $(k+2)$ edges $\{I_1, J_1\}, \{I_2, J_2\}, \ldots, \{I_l, J_l\}$, with $l \geq k+2$, $I_i \in \boldsymbol{I}$ and $J_i \in L$ for all $i \in [l]$, see a pair $\{x, y\}$ of different vertices, such that $x \in Z \cup B$ is adjacent to $I_i$, $y \in Z$ is adjacent to $J_i$ for all $i \in [l]$, then we add $\{x, y\}$ to the set of pair-constraints $\mathcal{P}$.

At first sight Reduction Rule 13.7 and 13.9 may seem somewhat similar, but on closer inspection one can observe a decisive difference. Obviously, both reduction rules use the fact that if there are $k+2$ disjoint $S$-edges seeing a pair $\{x, y\}$ of different vertices, then either $x$ or $y$ must be in a solution of size at most $k$; hence it is safe to add the pair $\{x, y\}$ to the set of pair-constraints. But, we need different arguments to show that the $k+2$ $S$-edges are disjoint. In Reduction Rule 13.7 it is clear that these $S$-edges are disjoint, because $M$ is a matching. This is not the case in Reduction Rule 13.9, where we have to use a different argument which follows from Lemma 13.11 (see Lemma 13.12).

The difference of these two reduction rules is that in Reduction Rule 13.9 we consider only edges between the two disjoint sets $\boldsymbol{I}$ and $L$. For this reason we can require in Reduction Rule 13.9 that every endpoint of the at least $k+2$ edges that is contained in $\boldsymbol{I}$ is adjacent to the vertex $x$ and every endpoint that is contained in $L$ is adjacent to vertex $y$. This is not possible in Reduction Rule 13.7, because an edge in the matching $M$ can be between any type of non-isolated bubbles.

**Reduction Rule 13.10.** If there exists an edge $e = \{I, J\}$ with $I \in \boldsymbol{I}$ and $J \in L$ such that $e$ sees no single vertex $z \in Z$ and for every pair $\{x, y\}$, with $x \in Z \cup B$, $y \in Z$, seen by $e$ the pair $\{x, y\}$ is a pair-constraint in $\mathcal{P}$, then we remove $e_S$ from $S$, delete $J$ from $L$ and replace $I$ by $I \cup J$ in $\boldsymbol{I}$. (Recall: $e_S$ is the unique edge in $E(V_I \cap V_J) \cap S$.)

Note that bubbles in $L$ are never adjacent to vertices in $B$, hence the vertex $z$ respectively the vertex $x$ cannot be contained in set $B$.

If we delete an edge $e_S \in E(V_I \cap V_J) \cap S$ from $S$ by applying Reduction Rule 13.10, then the consequence is that bubbles $I$ and $J$ are now merged into a single bubble. Note that it is sufficient to continue with Reduction Rule 13.9, because $M$ is still a matching that saturates all inner bubbles in the current graph $H_Z$ and $B$ still has the properties of Lemma 13.11 with respect to the current graph $H_{Z\cup B}$. That the edge set $M$ is still a matching in $H_Z$ holds because we never delete an edge in $M$ or an endpoint of an edge in $M$; we only merge an endpoint of an edge in $M$ with an exposed leaf bubble in $L$. The first two properties of Lemma 13.11 obviously hold with respect to the current graph $H_Z$. That property (iii) also holds follows from the fact that the leaf bubbles that are still in $L$ are the same as before and adjacent to the same inner bubbles as before.

**The reduction rules are safe.** First we show that our reduction rules are safe, i.e., that there exists a solution for $(G, S, \mathcal{P}, k)$ if and only if there exists a solution for $(G', S', \mathcal{P}', k')$. Obviously, Reduction Rules 13.1, 13.2, and 13.6 are safe. Furthermore, Reduction Rule 13.3 is safe because for every $S$-cycle $C$ through an edge $e \in S$ that is a bridge in $(V, E \setminus (S \setminus \{e\}))$ there is another $S$-edge $e' \neq e$ on the cycle $C$.

Let us consider the set $\mathcal{P}$ of pair-constraints to see that Reduction Rules 13.4 and 13.5 are safe. The set $\mathcal{P}$ naturally leads to the graph $P = (V(\mathcal{P}), \mathcal{P})$ and has the property that we have to pick at least one vertex of each pair-constraint for a solution for $(G, S, \mathcal{P}, k)$. Hence, any solution for $(G, S, \mathcal{P}, k)$ must contain a vertex cover of $P$. Thus, Reduction Rules 13.4 and 13.5 are direct analogues of classical reduction rules for the Vertex Cover problem, and therefore safe. To show that the other reduction rules are safe, we first show a technical lemma about a property of edges of graph $H_{Z\cup B}$.

**Lemma 13.12.** *If two different edges* $\{I_1, J_1\}$ *and* $\{I_2, J_2\}$ *in* $H_{Z\cup B}$ *with* $I_1, I_2 \in \boldsymbol{I}$, $J_1, J_2 \in L$ *see a single vertex* $z \in Z$ *or a pair* $\{x, y\}$ *with* $x \in Z \cup B$ *and* $y \in Z$ *such that* $\{z, I_1\}, \{z, I_2\}, \{z, J_1\}, \{z, J_2\} \in E(H^+_{Z\cup B})$ *or* $\{x, I_1\}, \{x, I_2\}, \{y, J_1\}, \{y, J_2\} \in E(H^+_{Z\cup B})$*, respectively, then it holds that they are disjoint, i.e., that* $I_1 \neq I_2$ *and* $J_1 \neq J_2$.

**Proof.** Note that if $J_1 = J_2$, then it holds that $I_1 = I_2$, because every leaf bubble in $L$ sees only one other bubble. Thus, to finish the proof it suffices to show that $I_1 \neq I_2$.

Assume for contradiction that $I_1 = I_2$. This implies that $J_1$ and $J_2$ are leaf bubbles in $L$ which are adjacent to the same inner bubble $I = I_1 = I_2$ in $H_{Z\cup B}$. For $J_1$ and $J_2$ it must hold that $z \in N_G(V_{J_i})$ or $y \in N_G(V_{J_i})$ for $i = 1, 2$. But this is a contradiction to property (iii) of Lemma 13.11. ■

To show that Reduction Rules 13.7 and 13.9 are safe, we have to prove that we only add a pair $\{x, y\}$ of vertices to the set $\mathcal{P}$ of pair-constraints if either $x$ or $y$ must be in each solution of size at most $k$. The $(k + 2)$ edges that see a pair $\{x, y\}$ are pairwise disjoint, because they are edges of matching $M$ or because Lemma 13.12 holds. Thus, we have at least $(k + 2)$ disjoint $x, y$-paths in $H^+_Z$ or $H^+_{Z\cup B}$ which we can extend to at least $(k + 2)$ disjoint $x, y$-paths in $G$ and each of these paths contains an $S$-edge. This is the reason why at least one of the vertices $x$ and $y$ must be in any solution of size at most $k$ (otherwise we have to delete at least $k + 1$ vertices, one of each path) and it is safe to add $\{x, y\}$ to the set $\mathcal{P}$ as a pair-constraint.

It remains to show that Reduction Rules 13.8 and 13.10 are safe. For this we prove that the $S$-edges that we delete in these reduction rules are irrelevant. First we prove the following lemma.

**Lemma 13.13.** *Let* $Y \subseteq V \setminus V(S)$ *be a superset of* $Z$*, hence* $G - Y$ *contains no* $S$*-cycle. If* $e = \{I, J\} \in H_Y$ *sees no single vertex* $y \in Y$ *and for every pair* $\{x, y\}$ *with* $x, y \in Y$ *seen by* $e$ *the pair* $\{x, y\}$ *is a pair-constraint in* $\mathcal{P}$*, then the unique edge* $e_S$ *in* $E(V_I, V_J) \cap S$ *is irrelevant for the instance* $(G, S, \mathcal{P}, k)$.

**Proof.** Let $e = \{I, J\} \in H_Y$ be an edge with the properties of the lemma and let $e_S = \{v_I, v_J\}$ be the single edge in $E(V_I, V_J) \cap S$. To show that $e_S$ is irrelevant for instance $(G, S, \mathcal{P}, k)$ we have to show that $X \subseteq V(G)$ is a solution for $(G, S, \mathcal{P}, k)$ if and only if $X$ is a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$. Since every solution $X$ for $(G, S, \mathcal{P}, k)$ is also a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$, we only have to show the other direction.

Let $X$ be a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$. We assume for the sake of contradiction that there exists an $S$-cycle $C$ in $G - X$. This $S$-cycle $C$ can only contain the $S$-edge $e_S$. Otherwise, $C$ would be an $(S \setminus \{e_S\})$-cycle which contradicts the assumption that $X$ is a solution for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$.

Claim 13.14. *If an $S$-cycle $C$ in $G$ only contains the $S$-edge $e_S$, then there exists either a vertex $y \in Y$ such that $e$ sees the single vertex $y$ and $y$ is contained in cycle $C$ or two different vertices $x, y \in Y$ such that $e$ sees the pair $\{x, y\}$ and cycle $C$ contains $x$ and $y$.*

Proof. Let $C$ be an $S$-cycle with the properties of the claim. Thus, $C$ must exit bubble $I$ and bubble $J$ by edges that end in $Y$, because this is the only way to obtain a path from $v_I$ to $v_J$ that uses no edge from $S$. If these two edges share their endpoint $y$ in $Y$, then $e$ sees the single vertex $y$ and $y$ is contained in $C$. On the other hand if these two edges have different endpoints $x, y$ in $Y$, then $e$ sees the pair $\{x, y\}$ and the vertices $x, y$ are contained in $C$. □

Based on the claim, it follows that edge $e = \{I, J\}$ must see a single vertex $y \in Y$ that is contained in $C$ or a pair $\{x, y\}$ with $x, y \in Y$ such that $x$ and $y$ are contained in $C$. However, edge $e$ sees no single vertex and every pair $\{x, y\}$ that is seen by $e$ must be contained in a pair-constraint of $\mathcal{P}$ (this follows from the properties of edge $e$ according to the requirements of the lemma).

Hence, the edge $e$ sees only pairs $\{x, y\}$ with $x, y \in Y$ that are contained in the set $\mathcal{P}$. Let $\{x, y\}$ be the pair that is seen by $e$ such that $x, y$ are vertices of cycle $C$ (using the claim). But at least one vertex of the pair $\{x, y\}$ must be in the solution $X$ for $(G, S \setminus \{e_S\}, \mathcal{P}, k)$. Since $e$ sees only pairs that are contained in the set $\mathcal{P}$ of pair-constraints, the cycle $C$ is no cycle in $G - X$. ■

By choosing $Y = Z$ or $Y = Z \cup B \subseteq Z$ it follows directly from Lemma 13.13 that we only delete an edge $e_S$ in Reduction Rule 13.8 or Reduction Rule 13.10, respectively, when the $S$-edge $e_S$ is irrelevant for instance $(G, S, \mathcal{P}, k)$.

**Applying the Reduction Rules.** First, we show that if none of the reduction rules is applicable, then the size of $S$ is bounded by $\mathcal{O}(k^4)$. For this we prove two lemmas. One bounds the size of $M$ which helps us to bound the number of inner bubbles as well as leaf bubbles that are not in $L$ and the other bounds the number of leaf bubbles in $L$.

**Lemma 13.15.** *If the matching $M$ saturates all inner bubbles in $H_Z$ and we cannot apply Reduction Rules 13.1 through 13.8, then the size of $M$ is at most $\mathcal{O}(k^3)$.*

**Proof.** Each edge in $M$ sees either a pair of vertices in $Z$ that does not form a pair-constraint in $\mathcal{P}$ or a single vertex in $Z$. Otherwise, we can apply Reduction Rules 13.2, 13.3 or 13.8. Recall that Reduction Rule 13.2 and Reduction Rule 13.3 ensure that each bubble is adjacent to a vertex in $Z$ and that Reduction Rule 13.8 deletes an edge $e$ from the matching $M$, or more precisely, deletes the edge in $S$, that corresponds to the matching edge, from the set $S$, when the edge $e$ sees neither a single vertex in $Z$ nor a pair $\{x, y\}$ with $x, y \in Z$ that is not contained in a pair-constraint of $\mathcal{P}$.

The number of pairs in $Z$ is at most $\binom{|Z|}{2} \leq |Z|^2$. Therefore, the number of pairs in $Z$ that are not in the set $\mathcal{P}$ of pair-constraints is at most $|Z|^2$. Because we cannot apply

Reduction Rule 13.7, at most $(k+1)$ edges in $M$ see any pair that is not in the set of pair-constraints. Thus, at most $(k+1)|Z|^2$ edges of $M$ can see a pair of vertices in $Z$ that is not in $\mathcal{P}$.

The number of edges in $M$ that see a single vertex in $Z$ is at most $k|Z|$. Otherwise, we can apply Reduction Rule 13.6, because at least one single vertex $z$ in $Z$ is seen by at least $k+1$ edges from $M$ and these edges together with $z$ are a $z$-flower of order $k+1$ in $H_Z^+$ which we can expand to a $z$-flower of order $k+1$ in $G$. Since we cannot apply Reduction Rules 13.6, 13.7 or 13.8, this leads to at most $(k+1)|Z|^2 + k|Z| \in \mathcal{O}(k^3)$ edges in $M$, because $|Z| \leq 8k$. ∎

From the Lemma 13.15 it follows that the number of inner bubbles in $H_Z$ is at most $2|M| \in \mathcal{O}(k^3)$.

**Lemma 13.16.** *If we cannot apply Reduction Rules 13.1 through 13.10 then the size of $L$ is bounded by $\mathcal{O}(k^4)$.*

**Proof.** We claim that the number of edges between bubbles in $\boldsymbol{I}$ and bubbles in $L$ is at most $(k+1)|Z|(|B|+|Z|) + k|Z|$, if no reduction rule is applicable. This implies that there are at most $\mathcal{O}(k^4)$ leaf bubbles in $L$, because $|Z| \leq 8k$ and $|B| \leq 2k|Z|$.

By Reduction Rule 13.10, each edge between a bubble $I$ in $\boldsymbol{I}$ and an exposed leaf bubble $J$ in $L$ sees a pair $\{x, y\}$, that is not contained in $\mathcal{P}$, meaning that $\{x, I\}, \{y, J\} \in E(H^+_{Z\cup B})$ for $x \in Z \cup B$, $y \in Z$, or sees a single vertex $z$ in $Z$. The number of pairs in $Z \times (Z \cup B)$ is at most $|Z|(|Z| + |B|)$.

Reduction Rule 13.9 adds a pair $\{x, y\}$ to the set $\mathcal{P}$ of pair-constraints if at least $(k+2)$ edges $\{I_1, J_1\}, \{I_2, J_2\}, \ldots, \{I_l, J_l\}$ with $l \geq k+2$, $I_i \in \boldsymbol{I}$ and $J_i \in L$ for all $i \in [l]$, see the pair $\{x, y\}$ such that $x \in Z \cup B$ is adjacent to $I_i$ and $y \in Z$ is adjacent to $J_i$ for all $i \in [l]$. This bounds the number of edges between vertices in $\boldsymbol{I}$ and $L$ which see a pair, whose vertices are not a pair in the set $\mathcal{P}$ of pair-constraints, by $(k+1)|Z|(|Z| + |B|)$. The number of edges between vertices in $\boldsymbol{I}$ and $L$ that see a certain vertex $z \in Z$ is at most $k$, otherwise the at least $k+1$ edges between $\boldsymbol{I}$ and $L$ that see vertex $z$ together with the vertex $z$ form a $z$-flower of order $k+1$ in $H^+_{Z\cup B}$, because Lemma 13.12 ensures that the edges are disjoint. But then we can apply Reduction Rule 13.6 and delete the vertex $z$. Hence at most $k|Z|$ edges between vertices $\boldsymbol{I}$ and $L$ can see a vertex in $Z$. This leads to at most $(k+1)|Z|(|B|+|Z|)+k|Z|$ edges between vertices in $\boldsymbol{I}$ and $L$, because we cannot apply Reduction Rules 13.6, 13.9 or 13.10. This implies that $|L| \in \mathcal{O}(k^4)$, because $|Z| \leq 8k$ and $|B| \leq 2k|Z| \leq 16k^2$. ∎

If we combine these two results, we know that $|\mathcal{D}_Z^i| + |\mathcal{D}_Z^l| \in \mathcal{O}(k^4)$. Recall that there is at most one edge of $S$ between any two bubbles and that $V(S) \cap Z = \emptyset$. Together with the fact that $H_Z$ is a forest, it follows that $|\mathcal{D}_Z^i| + |\mathcal{D}_Z^l|$ is an upper bound for the number of $S$-edges.

Finally we have to prove that we can perform the reduction in polynomial time. First we prove that each reduction rule is applied a polynomial number of times and second that every reduction rule application can be performed in polynomial time.

**Lemma 13.17.** *Each reduction rule is applied at most a polynomial number of times.*

**Proof.** Observe that we reduce in each reduction rule, except Reduction Rules 13.7 and 13.9, the size of at least one of the sets $V$, $E$, $S$, the value $k$ or decide that no solution of size at most $k$ exists. Furthermore, we never increase the size of such sets or the parameter $k$. In Reduction Rules 13.7 and 13.9 we add pair-constraints to $\mathcal{P}$, but if $\mathcal{P}$ contains more than $k^2$ pair-constraints, we either find a vertex $z \in V(\mathcal{P})$ that we delete in Reduction Rule 13.4 and reduce $k$ by one or we decide in Reduction Rule 13.5 that no solution of size at most $k$ exists. This bounds the number of pair-constraints that we add to $\mathcal{P}$ during the reduction by $k^3$ because we can decrease $k$ at most $k$ times. Thus, each reduction rule is applied at most a polynomial number of times. ∎

Next, we show that each reduction rule application can be performed in polynomial time. It is obvious that we can apply Reduction Rules 13.1 through 13.5 in polynomial time. The following lemma addresses Reduction Rule 13.6 by solving a matroid parity problem on an appropriate gammoid.

**Lemma 13.18.** *Let $G = (V, E)$, $z \in V$, and $S \subseteq E$. A $z$-flower of maximum order, i.e., a maximum number of $S$-cycles that intersect only in $z$, can be found in (deterministic) polynomial time.*

**Proof.** For simplicity, we assume that there are no edges of $S$ incident with $z$ and that no two edges of $S$ are incident with the same vertex of $G$. If this is not the case, then we subdivide every $S$-edge $e = \{v, w\}$ into three edges $\{v, v_e\}, \{v_e, v_w\}, \{v_w, w\}$. In the set $S$ we replace edge $e$ by the edge $\{v_e, w_e\}$. In the resulting graph no $S$-edge is incident with an original vertex in the graph or to an other $S$-edge. Subdividing the $S$-edges like this does not change the maximum order of $z$-flowers. Furthermore, the set of $z$-flowers does not change and it is easy to transfer a $z$-flower in the original graph to one in the new graph and vice versa.

Let $\{C_1, \ldots, C_t\}$ be a $z$-flower of order $t$. Each $C_i$ gives rise to a path $P_i$ between two different neighbors $u$ and $v$ of $z$; all these paths are fully vertex-disjoint. By our above assumption, there are no $S$-edges incident with $z$, hence, each $P_i$ must contain two consecutive vertices, say $s_i$ and $t_i$, with $\{s_i, t_i\} \in S$. In this way, each path $P_i$ can be split into two paths, $P_{i,s}$ and $P_{i,t}$, from $N(v)$ to $\{s_i, t_i\}$; all these $2t$ paths are pairwise vertex-disjoint and do not contain the vertex $z$. Thus, from any $z$-flower of order $t$ we get $2t$ vertex-disjoint paths in $G - z$ from $N(z)$ to $T \subseteq V(S)$, i.e., endpoints of $S$-edges, such that $T$ can be partitioned into $t$ two-sets of vertices that are also edges in $S$. In the language of gammoids this means that $T$ is an independent set in the gammoid on the graph $G - z$, with sources $N(z)$, and ground set $V(S)$.

Conversely, any independent set $T$ in the mentioned gammoid implies the existence of $|T|$ vertex-disjoint paths in $G - z$ from $N(z)$ to $T$. If, as above, $T$ can be partitioned into edges of $S$ then this gives rise to a $z$-flower of order $t = |T|/2$: Clearly, $|T|$ must be even to allow for the partition into sets of size two. Moreover, the paths are vertex-disjoint and, thus, two paths from $N(z)$ ending in $\{s_i, t_i\} \in S$ can be combined, using that $\{s_i, t_i\}$ must be an edge of $G$ into a single path, say $P_i$, from $N(z)$ to $N(z)$ that contains at least one edge of $S$. Note that, because $s_i$ and $t_i$ are endpoints of two paths in the packing they cannot occur in any other path, so this combination still yields vertex-disjoint paths in $G - z$. Finally, adding the vertex $z$, the paths $P_1, \ldots, P_t$ can be combined into $t$ $S$-cycles that intersect only in $z$.

Thus, the task of finding a $z$-flower of maximum order reduces to that of solving a matroid parity problem on a gammoid: The underlying graph is $G - z$, the source set is $N_G(z)$, the ground set is $V(S)$, and the pairs are given by $S$. Recall that pairs in $S$ are vertex-disjoint. Using the algorithm due to Lovász [Lov80], one may find a maximum independent set composed of pairs in $S$ in polynomial time, when provided with a matrix representation for the gammoid. A small caveat would be that one would need a randomized algorithm for finding said representation. Conveniently, specialized deterministic algorithms exist for subclasses of linear matroids; we can use a deterministic algorithm due to Tong et al. [TLV82] that solves the problem by reduction to weighted matching on graphs. (Note that given a maximum independent set $T$ composed of pairs, the cycles of the $z$-flower can be found by computing vertex-disjoint paths from $N(z)$ to $T$ in $G - z$ via a vertex-capacitated flow computation.) ■

It remains to show that we can apply Reduction Rules 13.7 through 13.10 in polynomial time.

**Lemma 13.19.** *We can apply Reduction Rule 13.7 and Reduction Rule 13.8 in polynomial time.*

**Proof.** First of all, we store for each edge $e = \{I, J\} \in M$ all vertices $z \in Z$ seen by edge $e$ and all pairs $\{x, y\}$ with $x, y \in Z$ seen by edge $e$. For each edge we need at most $\mathcal{O}(|Z|^2)$ time; we only have to test for each vertex $z \in Z$ and each pair $\{x, y\}$ with $x, y \in Z$ whether $\{I, z\}, \{J, z\} \in E(H_Z)$ respectively $\{I, x\}, \{J, y\} \in E(H_Z)$ or $\{I, y\}, \{J, x\} \in E(H_Z)$.

Next, we count how many edges see a pair $\{x, y\}$ with $x, y \in Z$ and denote this value by $c_{\{x,y\}}$. It takes at most $\mathcal{O}(|E||Z|^2)$ time to compute all values, because we only have to count for how many edges we store a certain pair. If a counter $c_{\{x,y\}}$ has value at least $k + 2$, then we add the pair $\{x, y\}$ to the set $\mathcal{P}$ of pair-constraints. We can check this for all counters in $\mathcal{O}(|Z|^2)$ time. The above computation corresponds to the computation we need for Reduction Rule 13.7.

To apply Reduction Rule 13.8 we only have to look at all vertices and pairs that we stored for an edge $e \in M$. If we have stored no single vertex and only pairs that are pair-constraints in $\mathcal{P}$, then $e$ fulfills the conditions of an edge that we delete in Reduction Rule 13.8. To check this for one edge takes time at most $\mathcal{O}(|Z|^2)$. ■

The proof that we can apply Reduction Rule 13.9 and Reduction Rule 13.10 in polynomial time is similar to the proof that we can apply Reduction Rule 13.7 and Reduction Rule 13.8 in polynomial time. The only difference is that we have to remember which endpoint of an edge is adjacent to which vertex in a pair.

**Lemma 13.20.** *We can apply Reduction Rule 13.9 and Reduction Rule 13.10 in polynomial time.*

**Proof.** As in the proof of Lemma 13.19, we store for each edge $e = \{I, J\}$ with $I \in \boldsymbol{I}$, $J \in L$ all vertices $z \in Z$ seen by edge $e$ and all pairs $(x, y)$ with $x \in Z \cup B$ adjacent to $I$, $y \in Z$ adjacent to $J$ such that $e$ sees the pair $\{x, y\}$. For each edge $e = \{I, J\}$ with $I \in \boldsymbol{I}$, $J \in L$ we need at most $\mathcal{O}(|Z \cup B||Z|)$ time: We only have to test for each vertex $z \in Z$ and each pair $(x, y)$ with $x \in Z \cup B$, $y \in Z$ whether $\{I, z\}, \{J, z\} \in E(H_{Z\cup B})$ or $\{I, x\}, \{J, y\} \in E(H_{Z\cup B})$, respectively. Afterwards, we count for how many edges we stored the pair $(x, y)$ with $x \in Z \cup B$, $y \in Z$ and denote this value by $c_{(x,y)}$. It takes at most $\mathcal{O}(|E||Z \cup B||Z|)$ time to compute all values. If a counter $c_{(x,y)}$ has value at least $k + 2$, then we add the pair $\{x, y\}$ to the set $\mathcal{P}$ of pair-constraints. We can check this for all counters in $\mathcal{O}(|Z \cup B||Z|)$ time. The above computation corresponds to the computation we need for Reduction Rule 13.9, because we only store the pair $(x, y)$ for an edge if the edge sees the pair $\{x, y\}$.

To apply Reduction Rule 13.10 we only have to consider all vertices and pairs that we stored for an edge $e$ between the bubbles in $\boldsymbol{I}$ and the bubbles in $L$. If we have stored no single vertex and only pairs $(x, y)$ such that $\{x, y\}$ is a pair-constraints in $\mathcal{P}$, then $e$ fulfills the conditions of an edge that we delete in Reduction Rule 13.10. To check this for one edge takes time at most $\mathcal{O}(|Z \cup B||Z|)$. ■

Finally, we show that we can compute the matching $M$ and the set $B$, which is the union of all $z$-blockers $B_z$ in $G_z$ with $z \in Z$, in polynomial time.

**Lemma 13.21.** *We can compute a maximal matching $M$ in $H_Z$ that saturates all inner bubbles in polynomial time.*

**Proof.** The graph $H_Z$ is a forest where all inner bubbles have degree greater than one. Thus, it is enough to show that every forest $F$ has a maximal matching that saturates all vertices of degree greater than one. We will prove this by induction on the number of edges.

If the forest $F$ has no edges, then $M = \emptyset$ is a valid solution that saturates all vertices of degree greater than one. Otherwise, since $F$ is a forest with at least one edge, there exists at least one vertex $v$ of degree one. By induction, $F' = F - v$ has a maximal matching $M'$ that saturates all vertices of degree greater than one. Now, if $M'$ saturates the unique neighbor $u$ of $v$, then $M = M'$ is a matching that saturates all vertices of degree greater than one in the forest $F$. Otherwise, the matching $M' = M \cup \{\{u, v\}\}$ is a matching that saturates all vertices of degree greater than one in the forest $F$.

This argument can be easily converted into a recursive algorithm for computing the desired matching in polynomial time. ■

It remains to show that we can find the set $B$ in polynomial time. From Schrijver's proof of Theorem 13.1 [Sch01] and the proof of Lemma 13.10 it follows that we can find in polynomial time either a $z$-flower of order $(k + 1)$ or a $z$-blocker of size at most $2k$

in $G_z$. Since there exists no $z$-flower in $G_z$ when we compute $B$, respectively the sets $B_z$ with $z \in Z$, we compute for every vertex $z \in Z$ exactly once the set $B_z$ and since $B$ is simply the union of all $z$-blockers we can compute $B$ in polynomial time.

**Finding an equivalent instance for Edge Subset FVS.** Up to now we can only bound the number of edges in $S$ for the PAIR-CONSTRAINED EDGE SUBSET FVS problem. As mentioned above the instance $(G, S, \mathcal{P} = \emptyset, k)$ for PAIR-CONSTRAINED EDGE SUBSET FVS is equivalent to the instance $(G, S, k)$ of EDGE SUBSET FVS. Therefore we only have to show that we can find in polynomial time an instance of EDGE SUBSET FVS that is equivalent to the instance $(G, S, \mathcal{P}, k)$ of PAIR-CONSTRAINED EDGE SUBSET FVS and has at most $\mathcal{O}(k^4)$ $S$-edges. Let $\{x, y\} \in \mathcal{P}$ be a pair-constraint. If there are two different edges $e_1, e_2$ between $x$ and $y$ of which at least one, without loss of generality $e_1$, is contained in $S$, then $x$ or $y$ must be in every solution, because $xe_1ye_2x$ is an $S$-cycle in $G$.

For this reason, the instance $(G', S' = S \cup \mathcal{P}, k)$ of EDGE SUBSET FVS is equivalent to the instance $(G, S, \mathcal{P}, k)$ of PAIR-CONSTRAINED EDGE SUBSET FVS, where $G'$ is created from $G$ by adding one edge $\{x, y\}$ between every two vertices $x$ and $y$ with $\{x, y\} \in \mathcal{P}$ when $\{x, y\} \notin E$ and by adding an edge $\{x, y\}$ between $x$ and $y$ that is also contained in $S'$. Thus, there are two edges between the vertices $x$ and $y$ with $\{x, y\} \in \mathcal{P}$ in graph $G'$ and we add exactly one of the two edges between $x$ and $y$ to $S'$. Because we cannot apply Reduction Rule 13.4 or 13.5 to $(G, S, \mathcal{P}, k)$, we know that $|\mathcal{P}| \leq k^2$. This leads to a bound of $|S| + |\mathcal{P}| \in \mathcal{O}(k^4)$ edges in $S'$ for the EDGE SUBSET FVS problem after the reduction.

Finally, we combine the results of Sections 13.3 and 13.4 to obtain a polynomial kernel for EDGE SUBSET FVS parameterized by $k$. Let us first make some comments about the reduction of the size of $S$ and the kernelization: For the reduction of the size of $S$ we use the fact that we can always find a solution that is disjoint from $T$. This only holds because we modified the graph accordingly. But since this is a correct reduction it holds that an input instance $(G, S, k)$ of EDGE SUBSET FVS has a solution if and only if the output instance $(G', S', k')$ of the reduction in Section 13.4 has a solution. Thus it is no problem that we consider dominant solutions for the kernelization in Section 13.3 and that the kernelization only guarantees the preservation of dominant solutions. Every instance $(G', S', k')$ has a dominant solution of size at most $k'$ when a solution of size at most $k'$ exists. Recall that $X$ is a dominant solution for $(G', S', k')$ if it has minimum size and contains a maximal number of vertices from $T'$ among solutions of minimum size. Hence, if $(G', S', k')$ has a solution then it has a dominant solution $X$. Let $(G'', S', k')$ be the output instance of the kernelization in Section 13.3. From Lemma 13.6 it follows that $(G', S', k')$ has a solution if and only if $(G'', S', k')$ has a solution.

Summarized, the reduction of the number of edges in $S$ to $\mathcal{O}(k^4)$ edges together with the kernelization to $\mathcal{O}(|S|^2 k)$ vertices for EDGE SUBSET FVS parameterized by $|S|$ and $k$, results in a kernelized instance with $\mathcal{O}(k^9)$ vertices for EDGE SUBSET FVS parameterized by $k$.

## 13.5. Summary

First, using the matroid based tools of Kratsch and Wahlström [KW12] we showed that Edge Subset FVS parameterized by the solution size and the number of $S$-edges has a randomized polynomial kernel with $\mathcal{O}(|S|^2 k)$ vertices. The error-probability can be made exponentially small without increasing the size of the kernel (cf. [KW12]). Second, we showed that one can reduce the number of $S$-edges. More specifically, we showed that we can reduce an instance $(G, S, k)$ of Edge Subset FVS in polynomial time to an equivalent instance $(G', S', k')$ of Edge Subset FVS such that $|S'| \in \mathcal{O}(k^4)$ and $k' \leq k$. Combining these two results, we have shown that Subset FVS parameterized by the solution size admits a randomized polynomial kernel with $\mathcal{O}(k^9)$ vertices which answers the question of Cygan et al. [CPPW13b] positively.

# CHAPTER 14

# CONCLUSION AND OPEN PROBLEMS

In this chapter we recap the results of this part and we will present some open problems. We started with an introduction to graph cut problems and feedback problems, like Feedback Vertex Set, Odd Cycle Transversal, Subset FVS and Multiway Cut. So far, the best tools to obtain polynomial kernelization algorithms for feedback problems and graph cut problems are the matroid based tools of Kratsch and Wahlström [KW12]. Using these tools it was shown that Odd Cycle Transversal and Deletable Terminal Multiway Cut parameterized by the solution size as well as Vertex Cover above LP and also above 2LP − MM have randomized polynomial kernels [Kra18, KW12, KW14]. In Chapter 13 we showed that the Subset FVS problem has a randomized polynomial kernelization using the matroid based tools of Kratsch and Wahlström [KW12], positively answering the question of Cygan et al. [CPPW13b]. Our kernelization algorithm consists of two steps. First, we reduced to an equivalent instance with at most $\mathcal{O}(k^4)$ $S$-edges (Section 13.4) and second, we apply the randomized polynomial kernel when parameterized by the solution size and the number of $S$-edges (Section 13.3). As in previous work [KW12] the error-probability can be made exponentially small without increasing the kernel size.

**Open problems.** Although the error-probability can be made exponentially small, it would of course be very interesting whether the use of randomization and/or matroids can be avoided. Furthermore, there is quite a gap between a kernel for Subset FVS with $\mathcal{O}(k^9)$ vertices and a lower bound of size $\mathcal{O}(k^{2-\varepsilon})$ that is inherited from Vertex Cover [DvM14], conditioned on non-collapse of the polynomial hierarchy.

Other open problems regarding existence of polynomial kernels, possibly amenable to the matroid based tools, are Multiway Cut, Group FVS and Directed FVS. There is also a directed version of Subset FVS, called Directed Subset Feedback Vertex Set, but it generalizes Directed FVS, whose kernel status has remained open for quite some time now.

As for Vertex Cover and Edge Dominating Set we could also ask whether there are interesting structural parameters for which Subset FVS has (randomized) polynomial kernels. As mentioned in Chapter 12, Majumdar and Raman [MR18] showed that Feedback Vertex Set parameterized by the size of a modulator to a pseudoforest or mock-$d$-forest has a polynomial kernel. Now, one can define an $S$-pseudoforest as a graph where each connected component has at most one $S$-cycle, and ask whether Subset FVS parameterized by the size of a modulator to an $S$-pseudoforest has a (randomized) polynomial kernel. Depending on the outcome, one can continue to ask whether Subset FVS parameterized by the size of a modulator to an $S$-mock-$d$-forest has a polynomial kernel, where an $S$-mock-forest is a graph where each vertex is contained in at most one $S$-cycle, and an $S$-mock-$d$-forest is an $S$-mock-forest where the number of $S$-cycles in each connected component is at most $d$.

# Part V.

# Conclusion

# CHAPTER 15 SUMMARY

In this thesis, we considered three NP-complete problems, namely Vertex Cover, Edge Dominating Set and Subset FVS, and we studied whether these problems have polynomial kernels for certain parameters. We started with our favorite problem Vertex Cover. Motivated by the large number of positive results regarding kernelization algorithms parameterized by the size of a modulator [BS19, FS16, JB13, KW12, MRS18], especially the results of Fomin and Strømme [FS16] and Bougeret and Sau [BS19], we tried to generalize known results for kernelization algorithms with respect to structural parameters that are a modulator to a given graph class $\mathcal{C}$.

One result showed that there is a relation between the size of the largest minimal blocking set of graphs in $\mathcal{C}$, where $\mathcal{C}$ is a graph class that is closed under disjoint union, and the size, respectively the existence, of a polynomial kernel for Vertex Cover when parameterized by the size of a $\mathcal{C}$-modulator (Theorem 4.1). More precisely, we showed that Vertex Cover parameterized by the size of a $\mathcal{C}$-modulator $X$ does not have a kernel of size $\mathcal{O}(|X|^{\beta_{\mathcal{C}}-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, where $\beta_{\mathcal{C}}$ is the largest minimal blocking set size of a graph in graph class $\mathcal{C}$. Furthermore, we showed (tight) upper bounds for the largest minimal blocking set size for many graph classes (Chapter 5) such as $d$-quasi-forests (Theorem 5.4) and graphs with elimination distance at most $d$ to a hereditary graph class $\mathcal{C}$ (Theorems 5.12 and 5.17). Using the upper bounds on the largest minimal blocking set size, we were able to obtain new kernelization results (Chapter 6). However, we also showed that bounded minimal blocking set size alone is not sufficient to obtain polynomial kernels (Theorem 4.2).

Next, we considered the Edge Dominating Set problem which is on the one hand similar to Vertex Cover and on the other hand very different. In a way, one can see Edge Dominating Set as a Vertex Cover with pair-constraints, i.e., we have a graph $G = (V, E)$, an integer $k$ and a set of pairs $P \subseteq \binom{V}{c}$ for an integer $c$ and want to find a set $P' \subseteq P$ of size at most $k$ such that $V(P')$ is a vertex cover of $G$. Now, the "classical" Vertex Cover is the same as Vertex Cover with pair-constraints where $c = 1$ and Edge Dominating Set is the same as Vertex Cover with pair-constraints where $c = 2$ and $P = E$. Surprisingly, we showed that Edge Dominating Set is much more complicated than Vertex Cover when it comes to polynomial kernelization algorithms for structural parameters.

We have seen, for example, that Vertex Cover still has a polynomial kernel when parameterized by the size of a $(\mathcal{C}_{forest}, d)$-modulator[1] (Theorem 6.11) whereas Edge Dominating Set has no polynomial kernel even when parameterized by the size of a modulator to a disjoint union of $P_3$'s unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ (Theorem 9.1). Furthermore, we can reduce every instance $(G, k, X)$ of Vertex Cover to an instance with $\mathcal{O}(|X|^c)$ vertices if $G - X$ contains only components of constant size at most $c$, but we proved that for Edge Dominating Set even constant-size components in $G - X$ behave in a nontrivial way with respect to kernelization (Theorem 10.4). Furthermore, it follows from the full classification (Theorem 10.4) that the graph class $\mathcal{C}_1$, consisting of all graphs where every vertex has degree at most one, is the unique maximal hereditary graph class for which Edge Dominating Set parameterized by the size of a $\mathcal{C}$-modulator has a polynomial kernel.

Although the existence of polynomial kernels for Vertex Cover and Edge Dominating Set parameterized by structural parameters is very different, all our kernelizations have one thing in common. To reduce the number of connected components in $G - X$, where $X$ is the given modulator, we construct a bipartite graph and apply the Hopcroft Karp algorithm. For Vertex Cover we used the bipartite graph to mark connected components that have conflicts with a subset $Z \subseteq X$, i.e., connected components where we have to use more vertices when $Z$ has an empty intersection with the solution, whereas for Edge Dominating Set we used the bipartite graph to mark connected components that allow us to cover a subset $Z \subseteq X$ without using $|Z|$ additional edges.

Finally, we considered the Subset FVS problem. We showed that Subset FVS has a randomized polynomial kernel using the matroid based tools of Kratsch and Wahlström [KW12]. This proves once more how useful these matroid based tools are when it comes to kernelization algorithms for graph cut problems and feedback problems. Recall that the kernelization algorithm contains two separate preprocessing routines. First, we apply the preprocessing routine that changes the structure of the input, more precisely, it reduces the number of $S$-edges (Section 13.4). Second, we apply the kernelization algorithm that reduces to $\mathcal{O}(k|S|^2)$ vertices (Section 13.3).

[1] Recall that given a graph $G$, a $(\mathcal{C}_{forest}, d)$-modulator $X \subseteq V(G)$ is a set of vertices such that the elimination distance (for each connected component) of $G - X$ to the class of forests is at most $d$.

# CHAPTER 16 OPEN PROBLEMS AND OUTLOOK

First, let us recall the most interesting open problems that arise from this thesis. Afterwards, we will give an outlook into further research directions, especially the use of parameterized complexity in practical applications.

**Open problems.** We want to highlight some of the open problems and further research directions that arise from this thesis.[1] Regarding VERTEX COVER, one of the most interesting questions is whether we can unify all existing polynomial kernels by a single parameter. Beyond that it would be nice to know the largest size of a minimal blocking set of a graph $G$ with $d = \textsc{vc}(G) - (2\textsc{lp}(G) - \textsc{mm}(G))$, for each fixed $d \in \mathbb{N}$. Could this bound be $2d+3$? Note that a bound for the largest minimal blocking set size of such graphs implies that the graph class $\mathcal{C}_{2\textsc{lp}-\textsc{mm}}$ is $f$-solid and together with Theorem 6.11 we would receive a polynomial kernel that generalizes all existing polynomial kernels.

One drawback of our kernelization algorithms is that we reduce an instance $(G, k, X)$ of VERTEX COVER to an equivalent instance $(G', k, X')$ such that $|X'| \in \mathcal{O}(|X|^c)$ for a constant $c$ and such that $G' - X'$ belongs to a graph class $\mathcal{C}$ for which we know that VERTEX COVER parameterized by the size of a $\mathcal{C}$-modulator has a polynomial kernel. Most likely, this is not the best possible kernelization algorithm because we ignore all the information that we obtained during the preprocessing algorithm, which reduces to an equivalent parameterized instance of VERTEX COVER for which we know that there exists a polynomial kernel. It would be interesting to obtain a polynomial kernel that does not use the known kernelization algorithms or to find a way to use the information that we figured out during the preprocessing.

For EDGE DOMINATING SET one of the most important open problems is to obtain a tight bound for the kernel size when parameterized by the solution size $k$. So far, the

[1]For a more detailed open problem discussion see Chapters 7, 11 and 14.

best known kernelization algorithm reduces to a graph with at most $\max\{\frac{1}{2}k^2 + \frac{7}{2}k, 6k\}$ vertices and at most $\frac{8}{27}k^3 + \mathcal{O}(k^2)$ edges [Hag12], whereas we only know that we cannot reduce to size $\mathcal{O}(k^{2-\varepsilon})$ for any $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Moreover, as further work it would be nice to know whether Edge Dominating Set parameterized by the size of a modulator $X$ to a disjoint union of paths of length $3\mathbb{N}_0 + 1$ has a polynomial kernel. Analogously to the proof of Theorem 9.4 it is possible to show that one can reduce a given instance $(G, k, X)$ in polynomial time to an equivalent instance $(G', k', X')$ where $G' - X'$ has at most $\mathcal{O}(|X|)$ connected components and $|X'| \leq |X|$. The remaining problem is to find a way to reduce the length of the paths in $G' - X'$.

Further research directions in the area of graph cut problems and feedback problems could be to find either different (deterministic) tools to obtain polynomial kernels for such problems or to find a way to avoid the randomness in the matroid based tools. However, the most important open problems, regarding graph cut problems and feedback problems, are, whether Multiway Cut and Directed FVS have polynomial kernels when parameterized by the solution size.

Another research direction is to consider other problems than Vertex Cover, Edge Dominating Set or Feedback Vertex Set under structural parameters. A possible candidate is the $d$-Hitting Set problem. It is well known that this problem is equivalent to the Vertex Cover problem in hypergraphs where every edge contains at most $d$ vertices. Thus, this problem generalizes Vertex Cover. There are two possible parameters. Given an instance $(U, \mathcal{F}, k)$ of $d$-Hitting Set one can choose as parameter a set $X \subseteq U$ such that the instance $(U \setminus X, \mathcal{F}')$ with $\mathcal{F}' = \{F \in \mathcal{F} \mid F \cap X = \emptyset\}$ corresponds to a hypertree. This parameter would be a lower bound for the solution size. The other possible parameter is the size of a set $X \subseteq U$ such that the instance $(U \setminus X, \mathcal{F}'')$ with $\mathcal{F}'' = \{F \setminus X \mid F \in \mathcal{F}\}$ is a hypertree. Observe that this parameter is independent of the solution size.

**Outlook.** The idea behind parameterized complexity is to take advantage of the structure of NP-hard problems to obtain efficient algorithms for real-world instances and that it will have a value to practical computation. However, over the years the focus of parameterized complexity turned more and more into primarily theoretical research. By using the $\mathcal{O}^*$ or $n^{\mathcal{O}(1)}$ notation, one ignores the polynomial dependency of the input instance. Besides, for kernelization algorithms we often do not compute the actual running time. There is no doubt that this is beneficial for the initial research of whether there exists an fpt-algorithm or a kernelization for a parameterized problem and also to reduce the exponential dependency of the running time or the kernel size, respectively. But, most of the time there is no effort in optimizing the polynomial factor, neither for fpt-algorithms nor for kernelization algorithms.[2]

Still, theoretical research is essential in order to make progress and to push the boundaries of the field itself. Especially for new problems, special cases and new parameters for well-known problems, theoretical research can show which parameters are interest-

[2]Exceptions to this are, for example, Lokshtanov et al. [LRS18] and Iwata et al. [IYY18].

ing in general, before trying to optimize algorithms. Additionally, while not seeming to be useful for application, negative results can help in practice by showing which efforts will be fruitless and can be discarded. Furthermore, for new techniques it is not always obvious what their practical impact is right away and to which problems they can be applied to. For example, the parameter treewidth turned out to be very useful in optimal structure sequence alignment to specify the consensus structure of an RNA family [SLM+05].

Nevertheless, one should not lose sight of the practical part. To maintain a deeper relationship between parameterized algorithms and practice the Parameterized Algorithms and Computational Experiments Challenge (PACE) was conceived. The first challenge was 2016 and motivated, for example, Iwata to two papers about Feedback Vertex Set where the focus was on the actual running time. In the first paper, Iwata [Iwa17] showed that Feedback Vertex Set parameterized by the solution size $k$ has a polynomial kernel with $2k^2 + k$ vertices and $4k^2$ edges and that one can compute the reduced instance in time $\mathcal{O}(k^4 m)$, which is linear for constant $k$. Hence, this kernelization algorithm is still practical for large instances if the parameter value $k$ is not too large. Using this kernelization algorithm and an LP-based branching he won the first place of Track B (Feedback Vertex Set) in the 1st PACE challenge [DHJ+16].

In the second paper, Iwata and Kobayashi [IK19] showed that a branching algorithm for Feedback Vertex Set parameterized by the solution size $k$ that was developed through the 1st PACE challenge and is empirically fast is also theoretically fast. More precisely, they showed that this algorithm runs in time $\mathcal{O}(3.460^k n)$. Note that this is the currently fastest algorithm for Feedback Vertex Set parameterized by the solution size. This paper was not only motivated by the 1st PACE challenge, but also from empirical evaluations by Kiljan and Pilipczuk [KP18]. Their work shows that the use of pruning techniques is much more important than the choice of the branching rules to obtain fast algorithms. Overall, branching techniques seem to be a good choice to obtain empirically fast algorithms. The top six submissions to this PACE challenge used branching algorithms. All these branching algorithms start with simple preprocessing to achieve, for example, that the graph has minimum degree three.

This shows that preprocessing is also very useful even when we do not reduce to an equivalent instance whose size is bounded by a function in the parameter. It can be used to obtain certain structures that allow us to, for example, obtain better branching algorithms or other kernelization algorithms. This strategy is well known for fpt-algorithms and was used, for example, by Lokshtanov et al. [LNR+14] to obtain an fpt-algorithm for Vertex Cover above lp and by Cygan et al. [CPPW13a] to obtain an fpt-algorithm for Multiway Cut above lp. Lokshtanov et al. [LNR+14] always reduce to a graph $G$ where the unique optimum solution to LP($G$) assigns value $\frac{1}{2}$ to all vertices and where for all independent sets $X \subseteq V(G)$ it holds that $|N(X)| - |X| > 2$. The fpt-algorithm of Cygan et al. [CPPW13a] reduces to a graph $G$ where the unique solution to the LP-relaxation assigns value $\frac{1}{2}$ to all neighbors of terminal vertices, zero to all other vertices, and where the neighborhood of each terminal vertex is the unique minimum cut separating this terminal from all other terminals.

Observe that our reduction of the number of $S$-edges fits in this way of preprocessing. Cygan et al. [CPPW13b] presented two algorithms for Edge Subset FVS. The first one is a branching algorithm that runs in time $2^{\mathcal{O}(k \log |S|)} \cdot n^{\mathcal{O}(1)}$ and the second one uses iterative compression and runs in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. During the compression step, they solve an instance of Disjoint Edge Subset FVS and they showed that one can in polynomial time either reduce to an equivalent instance where the number of $S$-edges is bounded by $\mathcal{O}(k^3)$ or to ignore this instance. Now, using our preprocessing algorithm to reduce the number of $S$-edges from Section 13.4, one can first reduce the number of $S$-edges and then apply the simpler $2^{\mathcal{O}(k \log |S|)} \cdot n^{\mathcal{O}(1)}$ algorithm for Edge Subset FVS which leads also to an algorithm that runs in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$.

In the Parts II and III we focused on structural parameters that are modulators to tractable graph classes. These kind of parameters can be of great interest in practice because they are embarrassingly parallelizable. For example, the most simple fpt-algorithm for Vertex Cover when parameterized by the size of a feedback vertex set $X$, first guesses the intersection of the solution with the set $X$ and checks whether this intersection is a vertex cover of $X$. Afterwards, we have to find a vertex cover in each connected component that is sound with this guess, i.e., the neighbors of all uncovered vertices in $X$ must be in the solution. This can obviously be done in parallel for each connected component.

Overall, it would be of interest to focus more on preprocessing algorithms that are useful in practice, i.e., preprocessing algorithms that run in linear or quadratic time and either reduce to a graph with a certain structure or to a small graph. Similar to linear-time fpt-algorithms, one could develop linear-time kernelization algorithms. Moreover, linear-time kernelization algorithms could also be interesting for problems that are solvable in polynomial time, but not linear-time. This matches a new research line called "FPT inside $\mathsf{P}$" (cf. [GMN17]). The idea is to assign a parameter $k$ to a polynomial-time solvable problem and to find an algorithm that solves the problem in linear time or at least faster than the best known algorithm, when the parameter $k$ is constant. Since big data becomes more and more important, the concept of "FPT inside $\mathsf{P}$" as well as linear-time preprocessing for such data sets is of interest.

Besides fpt-algorithms and kernelization algorithms, the concept of Turing kernels[3] could be of huge interest in practice, especially if we can design Turing kernels in such a way that we can solve many small instances in parallel. Hence, even if we know that a problem has a polynomial kernelization but the time to compute such a kernel is cubic or worse, a good way to obtain a practical algorithm is to try to obtain a linear-time (or quadratic-time) Turing kernel.

[3] A *Turing kernel* for a parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ is a polynomial-time algorithm that decides whether a given instance $(x, k) \in \Sigma^* \times \mathbb{N}$ belongs to $Q$, when given access to an oracle that solves instance $(x', k')$ of $Q$ with $|x'|, k' \leq f(k)$, for some computable function $f$; the size is $f$.

# BIBLIOGRAPHY

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[ACF+04] Faisal N. Abu-Khzam, Rebecca L. Collins, Michael R. Fellows, Michael A. Langston, W. Henry Suters, and Christopher T. Symons. Kernelization algorithms for the vertex cover problem: Theory and experiments. In Lars Arge, Giuseppe F. Italiano, and Robert Sedgewick, editors, *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics, New Orleans, LA, USA, January 10, 2004*, pages 62–69. SIAM, 2004.

[ACMM05] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. O(sqrt(log n)) approximation algorithms for min UnCut, min 2CNF deletion, and directed cut problems. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 573–581. ACM, 2005.

[Adl78] Leonard M. Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 75–83. IEEE Computer Society, 1978.

[AFLS07] Faisal N. Abu-Khzam, Michael R. Fellows, Michael A. Langston, and W. Henry Suters. Crown structures for vertex cover kernelization. *Theory Comput. Syst.*, 41(3):411–430, 2007.

[AFN04] Jochen Alber, Henning Fernau, and Rolf Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. *J. Algorithms*, 52(1):26–56, 2004.

[ALS91] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

[BBG00] Ann Becker, Reuven Bar-Yehuda, and Dan Geiger. Randomized algorithms for the loop cutset problem. *J. Artif. Intell. Res.*, 12:219–234, 2000.

[BD16] Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016.

[BD17] Jannis Bulian and Anuj Dawar. Fixed-parameter tractable distances to sparse graph classes. *Algorithmica*, 79(1):139–158, 2017.

[BDFH09] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[BDT18] Nicolas Bousquet, Jean Daligault, and Stéphan Thomassé. Multicut is FPT. *SIAM J. Comput.*, 47(1):166–207, 2018.

[BEF$^+$06] Kevin Burrage, Vladimir Estivill-Castro, Michael R. Fellows, Michael A. Langston, Shev Mac, and Frances A. Rosamond. The undirected feedback vertex set problem has a poly($k$) kernel. In *IWPEC 2006*, volume 4169 of *LNCS*, pages 192–202. Springer, 2006.

[BEG$^+$17] Benjamin Bergougnoux, Eduard Eiben, Robert Ganian, Sebastian Ordyniak, and M. S. Ramanujan. Towards a polynomial kernel for directed feedback vertex set. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, volume 83 of *LIPIcs*, pages 36:1–36:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[BF12] Daniel Binkele-Raible and Henning Fernau. Parameterized measure & conquer for problems with no small kernels. *Algorithmica*, 64(1):189–212, 2012.

[BFR98] R. Balasubramanian, Michael R. Fellows, and Venkatesh Raman. An improved fixed-parameter algorithm for vertex cover. *Inf. Process. Lett.*, 65(3):163–168, 1998.

[BG93] Jonathan F. Buss and Judy Goldsmith. Nondeterminism within P. *SIAM J. Comput.*, 22(3):560–572, 1993.

[BGL02] Endre Boros, Martin Charles Golumbic, and Vadim E. Levit. On the number of vertices belonging to all maximum stable sets of a graph. *Discrete Applied Mathematics*, 124(1-3):17–25, 2002.

[BJK14] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.

[Bod94] Hans L. Bodlaender. On disjoint cycles. *Int. J. Found. Comput. Sci.*, 5(1):59–68, 1994.

[Bod09] Hans L. Bodlaender. Kernelization: New upper and lower bound techniques. In Jianer Chen and Fedor V. Fomin, editors, *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, volume 5917 of *Lecture Notes in Computer Science*, pages 17–37. Springer, 2009.

[BS19] Marin Bougeret and Ignasi Sau. How much does a treedepth modulator help to obtain polynomial kernels beyond sparse graphs? *Algorithmica*, 81(10):4043–4068, 2019.

[BTY11] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

[Bul17] Jannis Bulian. Parameterized complexity of distances to sparse graph classes. Technical Report UCAM-CL-TR-903, University of Cambridge, Computer Laboratory, February 2017.

[BvD10] Hans L. Bodlaender and Thomas C. van Dijk. A cubic kernel for feedback vertex set and loop cutset. *Theory Comput. Syst.*, 46(3):566–597, 2010.

[BW06] Sergiy Butenko and Wilbert E. Wilhelm. Clique-detection models in computational biochemistry and genomics. *European Journal of Operational Research*, 173(1):1–17, 2006.

[CC06] Miroslav Chlebík and Janka Chlebíková. Approximation hardness of edge dominating set problems. *J. Comb. Optim.*, 11(3):279–290, 2006.

[CC08] Miroslav Chlebík and Janka Chlebíková. Crown reductions for the minimum weighted vertex cover problem. *Discrete Applied Mathematics*, 156(3):292–312, 2008.

[CCF14] Yixin Cao, Jianer Chen, and Jia-Hao Fan. An $\mathrm{O}^*(1.84^{\mathrm{k}})$ parameterized algorithm for the multiterminal cut problem. *Inf. Process. Lett.*, 114(4):167–173, 2014.

[CCHM15] Rajesh Hemant Chitnis, Marek Cygan, Mohammad Taghi Hajiaghayi, and Dániel Marx. Directed subset feedback vertex set is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(4):28:1–28:28, 2015.

[CCL15] Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: New measure and new structures. *Algorithmica*, 73(1):63–86, 2015.

[CDR+03] James Cheetham, Frank K. H. A. Dehne, Andrew Rau-Chaplin, Ulrike Stege, and Peter J. Taillon. Solving large FPT problems on coarse-grained parallel machines. *J. Comput. Syst. Sci.*, 67(4):691–706, 2003.

[CFK+15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[CFL+08] Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.*, 74(7):1188–1198, 2008.

[CG05] L. Sunil Chandran and Fabrizio Grandoni. Refined memorization for vertex cover. *Inf. Process. Lett.*, 93(3):123–131, 2005.

[CHM13] Rajesh Hemant Chitnis, Mohammad Taghi Hajiaghayi, and Dániel Marx. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. *SIAM J. Comput.*, 42(4):1674–1696, 2013.

[CJ03] Liming Cai and David W. Juedes. On the existence of subexponential parameterized algorithms. *J. Comput. Syst. Sci.*, 67(4):789–807, 2003.

[CKJ01] Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *J. Algorithms*, 41(2):280–301, 2001.

[CKP+14] Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Clique cover and graph separation: New incompressibility results. *TOCT*, 6(2):6:1–6:19, 2014.

[CKX10] Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010.

[CLJ00] Jianer Chen, Lihua Liu, and Weijia Jia. Improvement on vertex cover for low-degree graphs. *Networks*, 35(4):253–259, 2000.

[CLL+08] Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.

[CLL09a] Jean Cardinal, Stefan Langerman, and Eythan Levy. Improved approximation bounds for edge dominating set in dense graphs. *Theor. Comput. Sci.*, 410(8-10):949–957, 2009.

[CLL09b] Jianer Chen, Yang Liu, and Songjian Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.

[CLP+14] Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On the hardness of losing width. *Theory Comput. Syst.*, 54(1):73–82, 2014.

[CNP+11] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS 2011*, pages 150–159. IEEE Computer Society, 2011.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971.

[CPP16] Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. On group feedback vertex set parameterized by the size of the cutset. *Algorithmica*, 74(2):630–642, 2016.

[CPPW13a] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3:1–3:11, 2013.

[CPPW13b] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM J. Discrete Math.*, 27(1):290–309, 2013.

[CS93] Jason Cong and M'Lissa Smith. A parallel bottom-up clustering algorithm with applications to circuit partitioning in VLSI design. In Alfred E. Dunlop, editor, *Proceedings of the 30th Design Automation Conference. Dallas, Texas, USA, June 14-18, 1993.*, pages 755–760. ACM Press, 1993.

[Cun89] William H. Cunningham. The optimal multiterminal cut problem. In Fred Roberts, Frank Hwang, and Clyde L. Monma, editors, *Reliability Of Computer And Communication Networks, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, December 2-4, 1989*, volume 5 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 105–120. DIMACS/AMS, 1989.

[DF92a] Rodney G. Downey and Michael R. Fellows. Fixed-parameter intractability. In *Proceedings of the Seventh Annual Structure in Complexity Theory Conference, Boston, Massachusetts, USA, June 22-25, 1992*, pages 36–49. IEEE Computer Society, 1992.

[DF92b] Rodney G. Downey and Michael R. Fellows. Fixed parameter tractability and completeness. In Klaus Ambos-Spies, Steven Homer, and Uwe Schöning, editors, *Complexity Theory: Current Research, Dagstuhl Workshop, February 2-8, 1992*, pages 191–225. Cambridge University Press, 1992.

[DF95] Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: on completeness for W[1]. *Theor. Comput. Sci.*, 141(1&2):109–131, 1995.

[DF99] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[DFD04] Rodney G. Downey, Michael R. Fellows, and Frank K. H. A. Dehne, editors. *Parameterized and Exact Computation, First International Workshop, IWPEC 2004, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3162 of *Lecture Notes in Computer Science*. Springer, 2004.

[DFL+07] Frank K. H. A. Dehne, Michael R. Fellows, Michael A. Langston, Frances A. Rosamond, and Kim Stevens. An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. *Theory Comput. Syst.*, 41(3):479–492, 2007.

[DFRS04] Frank K. H. A. Dehne, Michael R. Fellows, Frances A. Rosamond, and Peter Shaw. Greedy localization, iterative compression, modeled crown reductions: New FPT techniques, an improved algorithm for set splitting, and a novel 2k kernelization for vertex cover. In Downey et al. [DFD04], pages 271–280.

[DHJ+16] Holger Dell, Thore Husfeldt, Bart M. P. Jansen, Petteri Kaski, Christian Komusiewicz, and Frances A. Rosamond. The first parameterized algorithms and computational experiments challenge. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63 of *LIPIcs*, pages 30:1–30:9. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[Die12] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[DJP+94] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[DM12] Holger Dell and Dániel Marx. Kernelization of packing problems. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM*

*Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 68–81. SIAM, 2012.

[DvM14] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

[DW96] Geoffrey M Downs and Peter Willett. Similarity searching in databases of chemical structures. *Reviews in computational chemistry*, 7:1–66, 1996.

[EMPX15] Bruno Escoffier, Jérôme Monnot, Vangelis Th. Paschos, and Mingyu Xiao. New results on polynomial inapproximabilityand fixed parameter approximability of edge dominating set. *Theory Comput. Syst.*, 56(2):330–346, 2015.

[ENSZ00] Guy Even, Joseph Naor, Baruch Schieber, and Leonid Zosin. Approximating minimum subset feedback sets in undirected graphs with applications. *SIAM J. Discrete Math.*, 13(2):255–267, 2000.

[ENZ00] Guy Even, Joseph Naor, and Leonid Zosin. An 8-approximation algorithm for the subset feedback vertex set problem. *SIAM J. Comput.*, 30(4):1231–1252, 2000.

[Fel03] Michael R. Fellows. Blow-ups, win/win's, and crown rules: Some new directions in FPT. In Hans L. Bodlaender, editor, *Graph-Theoretic Concepts in Computer Science, 29th International Workshop, WG 2003, Elspeet, The Netherlands, June 19-21, 2003, Revised Papers*, volume 2880 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2003.

[Fer06] Henning Fernau. Edge dominating set: Efficient enumeration-based exact algorithms. In Hans L. Bodlaender and Michael A. Langston, editors, *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, volume 4169 of *Lecture Notes in Computer Science*, pages 142–153. Springer, 2006.

[FF56] Lester R. Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[FF62] Lester R. Ford and Delbert R. Fulkerson. *Flows in networks.* Princeton University Press, 1962.

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

[FGSS09] Fedor V. Fomin, Serge Gaspers, Saket Saurabh, and Alexey A. Stepanov. On two techniques of combining branching and treewidth. *Algorithmica*, 54(2):181–207, 2009.

[FHRV09] Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.

[FJR13] Michael R. Fellows, Bart M. P. Jansen, and Frances A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *Eur. J. Comb.*, 34(3):541–566, 2013.

[FLS14] Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In *SODA 2014*, pages 142–151. SIAM, 2014.

[FN02] Toshihiro Fujito and Hiroshi Nagamochi. A 2-approximation algorithm for the minimum weight edge dominating set problem. *Discrete Applied Mathematics*, 118(3):199–207, 2002.

[FS11] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct pcps for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[FS16] Fedor V. Fomin and Torstein J. F. Strømme. Vertex cover structural parameterization revisited. In *Graph-Theoretic Concepts in Computer Science - 42nd International Workshop, WG 2016, Istanbul, Turkey, June 22-24, 2016, Revised Selected Papers*, volume 9941 of *Lecture Notes in Computer Science*, pages 171–182, 2016.

[Gal61] Tibor Gallai. Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen. *Acta Mathematica Hungarica*, 12(1-2):131–173, 1961.

[Gal64] Tibor Gallai. Maximale Systeme unabhängiger Kanten. *Magyar Tud. Akad. Mat. Kutato Int. Kozl.*, 9:401–413, 1964.

[GG18] Hang Gao and Wenyu Gao. Maximum matching and kernelization of edge dominating set. *Inf. Process. Lett.*, 136:21–24, 2018.

[GGH$^+$06] Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. Syst. Sci.*, 72(8):1386–1396, 2006.

[GHKV15] Petr A. Golovach, Pinar Heggernes, Dieter Kratsch, and Yngve Villanger. An incremental polynomial time algorithm to enumerate all minimal edge dominating sets. *Algorithmica*, 72(3):836–859, 2015.

[GJ77] Michael R. Garey and David S. Johnson. The rectilinear steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[GKK14] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient classification for metric data. *IEEE Trans. Information Theory*, 60(9):5750–5759, 2014.

[GMN09] Jiong Guo, Hannes Moser, and Rolf Niedermeier. Iterative compression for exactly solving NP-hard minimization problems. In Jürgen Lerner, Dorothea Wagner, and Katharina Anna Zweig, editors, *Algorithmics of Large and Complex Networks - Design, Analysis, and Simulation [DFG priority program 1126]*, volume 5515 of *Lecture Notes in Computer Science*, pages 65–80. Springer, 2009.

[GMN17] Archontia C. Giannopoulou, George B. Mertzios, and Rolf Niedermeier. Polynomial fixed-parameter algorithms: A case study for longest path on interval graphs. *Theor. Comput. Sci.*, 689:67–95, 2017.

[GP16] Shivam Garg and Geevarghese Philip. Raising the bar for vertex cover: Fixed-parameter tractability above a higher guarantee. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1152–1166. SIAM, 2016.

[Gui11] Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011.

[GVY04] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.

[Hag12] Torben Hagerup. Kernels for edge dominating set: Simpler or smaller. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012 - 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings*, volume 7464 of *Lecture Notes in Computer Science*, pages 491–502. Springer, 2012.

[Hal35] Philip Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935.

[HHS82] Peter L. Hammer, Pierre Hansen, and Bruno Simeone. Vertices belonging to all or to no maximum stable sets of a graph. *SIAM Journal on Algebraic Discrete Methods*, 3(4):511–522, 1982.

[HK73] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.

[HK93] Joseph Douglas Horton and Kyriakos Kilakos. Minimum edge dominating sets. *SIAM J. Discrete Math.*, 6(3):375–387, 1993.

[HK17] Eva-Maria C. Hols and Stefan Kratsch. Smaller parameters for vertex cover kernelization. In Daniel Lokshtanov and Naomi Nishimura, editors, *12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria*, volume 89 of *LIPIcs*, pages 20:1–20:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[HK18] Eva-Maria C. Hols and Stefan Kratsch. A randomized polynomial kernel for subset feedback vertex set. *Theory Comput. Syst.*, 62(1):63–92, 2018.

[HK19] Eva-Maria C. Hols and Stefan Kratsch. On kernelization for edge dominating set under structural parameters. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 36:1–36:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

[HKP20] Eva-Maria C. Hols, Stefan Kratsch, and Astrid Pieterse. Elimination distances, blocking sets, and kernels for vertex cover. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, volume 154 of *LIPIcs*, pages 36:1–36:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[HP00] Ilker Hamzaoglu and Janak H. Patel. Test set compaction algorithms for combinational circuits. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 19(8):957–963, 2000.

[IK19] Yoichi Iwata and Yusuke Kobayashi. Improved analysis of highest-degree branching for feedback vertex set. *CoRR*, abs/1905.12233, 2019.

[IN16] Ken Iwaide and Hiroshi Nagamochi. An improved algorithm for parameterized edge dominating set problem. *J. Graph Algorithms Appl.*, 20(1):23–58, 2016.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[Iwa17] Yoichi Iwata. Linear-time kernelization for feedback vertex set. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of

*LIPIcs*, pages 68:1–68:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[IWY16] Yoichi Iwata, Magnus Wahlström, and Yuichi Yoshida. Half-integrality, LP-branching, and FPT algorithms. *SIAM J. Comput.*, 45(4):1377–1411, 2016.

[IYY18] Yoichi Iwata, Yutaro Yamaguchi, and Yuichi Yoshida. 0/1/all csps, half-integral a-path packing, and linear-time FPT algorithms. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 462–473. IEEE Computer Society, 2018.

[Jan13] Bart M. P. Jansen. *The power of data reduction: kernels for fundamental graph problems*. PhD thesis, Utrecht University, 2013.

[JB13] Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013.

[JP18] Bart M. P. Jansen and Astrid Pieterse. Polynomial kernels for hitting forbidden minors under structural parameterizations. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPIcs*, pages 48:1–48:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[JRV14] Bart M. P. Jansen, Venkatesh Raman, and Martin Vatshelle. Parameter ecology for feedback vertex set. *Tsinghua Science and Technology*, 19(4):387–409, 2014.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

[KK12] Ken-ichi Kawarabayashi and Yusuke Kobayashi. Fixed-parameter tractability for the subset feedback set problem and the S-cycle packing problem. *J. Comb. Theory, Ser. B*, 102(4):1020–1034, 2012.

[KLM+15] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, Lhouari Nourine, and Takeaki Uno. Polynomial delay algorithm for listing minimal edge dominating sets in graphs. In Frank Dehne, Jörg-Rüdiger Sack, and Ulrike Stege, editors, *Algorithms and Data Structures - 14th International Symposium, WADS 2015, Victoria, BC, Canada, August 5-7, 2015. Proceedings*, volume 9214 of *Lecture Notes in Computer Science*, pages 446–457. Springer, 2015.

[KLMN12] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine. On the neighbourhood Helly of some graph classes and applications to the enumeration of minimal dominating sets. In Kun-Mao Chao, Tsan-sheng Hsu, and Der-Tsai Lee, editors, *Algorithms and Computation - 23rd International Symposium, ISAAC 2012, Taipei, Taiwan, December 19-21, 2012. Proceedings*, volume 7676 of *Lecture Notes in Computer Science*, pages 289–298. Springer, 2012.

[KLW96] Ina Koch, Thomas Lengauer, and Egon Wanke. An algorithm for finding maximal common subtopologies in a set of protein structures. *Journal of Computational Biology*, 3(2):289–306, 1996.

[Ko82] Ker-I Ko. Some observations on the probabilistic algorithms and NP-hard problems. *Inf. Process. Lett.*, 14(1):39–43, 1982.

[Kön31] Dénes König. Graphs and matrices. *Math. Fiz. Lapok*, 38:116–119, 1931.

[Koz92] Dexter Campbell Kozen. *Design and Analysis of Algorithms.* Texts and Monographs in Computer Science. Springer, 1992.

[KP14] Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Inf. Process. Lett.*, 114(10):556–560, 2014.

[KP18] Krzysztof Kiljan and Marcin Pilipczuk. Experimental evaluation of parameterized algorithms for feedback vertex set. In Gianlorenzo D'Angelo, editor, *17th International Symposium on Experimental Algorithms, SEA 2018, June 27-29, 2018, L'Aquila, Italy*, volume 103 of *LIPIcs*, pages 12:1–12:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[KPPW15] Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Fixed-parameter tractability of multicut in directed acyclic graphs. *SIAM J. Discrete Math.*, 29(1):122–144, 2015.

[KPS04] Iyad A. Kanj, Michael J. Pelsmajer, and Marcus Schaefer. Parameterized algorithms for feedback vertex set. In Downey et al. [DFD04], pages 235–247.

[KPS08] Graham Kendall, Andrew J. Parkes, and Kristian Spoerer. A survey of NP-complete puzzles. *ICGA Journal*, 31(1):13–34, 2008.

[KR03] Daniel Kobler and Udi Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003.

[KR08] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[Kra18] Stefan Kratsch. A randomized polynomial kernelization for vertex cover with a smaller parameter. *SIAM J. Discrete Math.*, 32(3):1806–1839, 2018.

[KV12] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms.* Springer, Berlin, Heidelberg, 5th edition, 2012.

[KW12] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 450–459. IEEE Computer Society, 2012.

[KW14] Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms*, 10(4):20:1–20:15, 2014.

[Lam11] Michael Lampis. A kernel of order 2k-c log k for vertex cover. *Inf. Process. Lett.*, 111(23-24):1089–1091, 2011.

[LG07] Andrew R Leach and Valerie J Gillet. *An introduction to chemoinformatics.* Springer Science & Business Media, 2007.

[LN19] Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time. *CoRR*, abs/1906.12298, 2019.

[LNR$^+$14] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014.

[Lov77] László Lovász. Flats in matroids and geometric graphs. *Combinatorial surveys*, pages 45–86, 1977.

[Lov80] László Lovász. Matroid matching and some applications. *J. Comb. Theory, Ser. B*, 28(2):208–236, 1980.

[LP09] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[LRS16] Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. A linear time parameterized algorithm for directed feedback vertex set. *CoRR*, abs/1609.04347, 2016.

[LRS18] Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. Linear time parameterized algorithms for subset feedback vertex set. *ACM Trans. Algorithms*, 14(1):7:1–7:37, 2018.

[Mar06] Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.

[Mar09] Dániel Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009.

[MOR13] Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Trans. Algorithms*, 9(4):30:1–30:35, 2013.

[MR14] Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. *SIAM J. Comput.*, 43(2):355–388, 2014.

[MR18] Diptapriyo Majumdar and Venkatesh Raman. Structural parameterizations of undirected feedback vertex set: FPT algorithms and kernelization. *Algorithmica*, 80(9):2683–2724, 2018.

[MRS+11] Sounaka Mishra, Venkatesh Raman, Saket Saurabh, Somnath Sikdar, and C. R. Subramanian. The complexity of könig subgraph problems and above-guarantee vertex cover. *Algorithmica*, 61(4):857–881, 2011.

[MRS18] Diptapriyo Majumdar, Venkatesh Raman, and Saket Saurabh. Polynomial kernels for vertex cover parameterized by small degree modulators. *Theory Comput. Syst.*, 62(8):1910–1951, 2018.

[Nie04] Rolf Niedermeier. Ubiquitous parameterization - invitation to fixed-parameter algorithms. In Jirí Fiala, Václav Koubek, and Jan Kratochvíl, editors, *Mathematical Foundations of Computer Science 2004, 29th International Symposium, MFCS 2004, Prague, Czech Republic, August 22-27, 2004, Proceedings*, volume 3153 of *Lecture Notes in Computer Science*, pages 84–103. Springer, 2004.

[NR99] Rolf Niedermeier and Peter Rossmanith. Upper bounds for vertex cover further improved. In Christoph Meinel and Sophie Tison, editors, *STACS 99, 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, March 4-6, 1999, Proceedings*, volume 1563 of *Lecture Notes in Computer Science*, pages 561–570. Springer, 1999.

[NR00] Rolf Niedermeier and Peter Rossmanith. A general method to speed up fixed-parameter-tractable algorithms. *Information Processing Letters*, 73(3-4):125–129, 2000.

[NRRS12] N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. LP can be a cure for parameterized problems. In Christoph Dürr and Thomas Wilke, editors, *29th International Symposium on Theoretical*

*Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, volume 14 of *LIPIcs*, pages 338–349. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[NT75] George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8(1):232–248, 1975.

[Per68] Hazel Perfect. Applications of Menger's graph theorem. *J. Math. Anal. Appl.*, 22:96–111, 1968.

[Pri05] Elena Prieto. *Systematic kernelization in FPT algorithm design.* PhD thesis, The University of Newcastle, Australia, 2005.

[PW18] Marcin Pilipczuk and Magnus Wahlström. Directed multicut is W[1]-hard, even for four terminal pairs. *TOCT*, 10(3):13:1–13:18, 2018.

[Raz11] Igor Razgon. Large isolating cuts shrink the multiway cut. *CoRR*, abs/1104.5361, 2011.

[RK00] Chantal Roth-Korostensky. *Algorithms for building multiple sequence alignments and evolutionary trees.* PhD thesis, ETH Zurich, 2000. Diss. Technische Wissenschaften ETH Zürich, Nr. 13550, 2000.

[RO09] Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.

[RRS11] Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Paths, flowers and vertex cover. In Camil Demetrescu and Magnús M. Halldórsson, editors, *Algorithms - ESA 2011 - 19th Annual European Symposium, Saarbrücken, Germany, September 5-9, 2011. Proceedings*, volume 6942 of *Lecture Notes in Computer Science*, pages 382–393. Springer, 2011.

[RS95] Neil Robertson and Paul D. Seymour. Graph minors. XIII. the disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995.

[RS04] Neil Robertson and Paul D. Seymour. Graph minors. XX. Wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004.

[RSS06] Venkatesh Raman, Saket Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for finding feedback vertex sets. *ACM Trans. Algorithms*, 2(3):403–415, 2006.

[RSS07] Venkatesh Raman, Saket Saurabh, and Somnath Sikdar. Efficient exact algorithms through enumerating maximal independent sets and other techniques. *Theory Comput. Syst.*, 41(3):563–587, 2007.

[RSV04] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.

[SBS05] Dawn M. Strickland, Earl R. Barnes, and Joel S. Sokol. Optimal protein structure alignment using maximum cliques. *Operations Research*, 53(3):389–402, 2005.

[Sch01] Alexander Schrijver. A short proof of Mader's sigma-paths theorem. *J. Comb. Theory, Ser. B*, 82(2):319–321, 2001.

[SF99] Ulrike Stege and Michael Ralph Fellows. An improved fixed parameter tractable algorithm for vertex cover. *Technical report/Departement Informatik, ETH Zürich*, 318, 1999.

[SLM+05] Yinglei Song, Chunmei Liu, Russell L. Malmberg, Fangfang Pan, and Liming Cai. Tree decomposition based fast search of RNA structures including pseudoknots in genomes. In *Fourth International IEEE Computer Society Computational Systems Bioinformatics Conference, CSB 2005, Stanford, CA, USA, August 8-11, 2005*, pages 223–234. IEEE Computer Society, 2005.

[Ste99] Ulrike Stege. *Resolving conflicts in problems from computational biology.* PhD thesis, ETH Zurich, 1999. Diss. Technische Wissenschaften ETH Zürich, Nr. 13364, 2000.

[Sto73] Larry Stockmeyer. Planar 3-colorability is polynomial complete. *ACM Sigact News*, 5(3):19–25, 1973.

[SV12] Richard Schmied and Claus Viehmann. Approximating edge dominating set in dense graphs. *Theor. Comput. Sci.*, 414(1):92–99, 2012.

[SY11] Arezou Soleimanfallah and Anders Yeo. A kernel of order 2k-c for vertex cover. *Discrete Mathematics*, 311(10-11):892–895, 2011.

[Tho10] Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.

[TLV82] Po Tong, Eugene L. Lawler, and Vijay V. Vazirani. Solving the weighted parity problem for gammoids by reduction to graphic matching. In *Progress in Combinatorial Optimization*, pages 363–374. Academic Press, 1982.

[Tsu18] Dekel Tsur. Above guarantee parameterization for vertex cover on graphs with maximum degree 4. *CoRR*, abs/1812.10808, 2018.

[vRB12] Johan M. M. van Rooij and Hans L. Bodlaender. Exact algorithms for edge domination. *Algorithmica*, 64(4):535–563, 2012.

[VV86] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.

[Wah14] Magnus Wahlström. Half-integrality, LP-branching and FPT algorithms. In *SODA 2014*, pages 1762–1781. SIAM, 2014.

[Wah17] Magnus Wahlström. LP-branching algorithms based on biased graphs. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1559–1570. SIAM, 2017.

[WCFC09] Jianxin Wang, Beiwei Chen, Qilong Feng, and Jianer Chen. An efficient fixed-parameter enumeration algorithm for weighted edge dominating set. In Xiaotie Deng, John E. Hopcroft, and Jinyun Xue, editors, *Frontiers in Algorithmics, Third International Workshop, FAW 2009, Hefei, China, June 20-23, 2009. Proceedings*, volume 5598 of *Lecture Notes in Computer Science*, pages 237–250. Springer, 2009.

[Xia10a] Mingyu Xiao. Exact and parameterized algorithms for edge dominating set in 3-degree graphs. In Weili Wu and Ovidiu Daescu, editors, *Combinatorial Optimization and Applications - 4th International Conference, COCOA 2010, Kailua-Kona, HI, USA, December 18-20, 2010, Proceedings, Part II*, volume 6509 of *Lecture Notes in Computer Science*, pages 387–400. Springer, 2010.

[Xia10b] Mingyu Xiao. A note on vertex cover in graphs with maximum degree 3. In My T. Thai and Sartaj Sahni, editors, *Computing and Combinatorics, 16th Annual International Conference, COCOON 2010, Nha Trang, Vietnam, July 19-21, 2010. Proceedings*, volume 6196 of *Lecture Notes in Computer Science*, pages 150–159. Springer, 2010.

[Xia10c] Mingyu Xiao. Simple and improved parameterized algorithms for multiterminal cuts. *Theory Comput. Syst.*, 46(4):723–736, 2010.

[XKP13] Mingyu Xiao, Ton Kloks, and Sheung-Hung Poon. New parameterized algorithms for the edge dominating set problem. *Theor. Comput. Sci.*, 511:147–158, 2013.

[XN13] Mingyu Xiao and Hiroshi Nagamochi. Parameterized edge dominating set in graphs with degree bounded by 3. *Theor. Comput. Sci.*, 508:2–15, 2013.

[XN14] Mingyu Xiao and Hiroshi Nagamochi. A refined exact algorithm for edge dominating set. *Theor. Comput. Sci.*, 560:207–216, 2014.

[Yap83] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.

[YG80] Mihalis Yannakakis and Fanica Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.

# SELBSTÄNDIGKEITSERKLÄRUNG

Ich erkläre, dass ich die Dissertation selbständig und nur unter Verwendung der von mir gemäß § 7 Abs. 3 der Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät, veröffentlicht im Amtlichen Mitteilungsblatt der Humboldt-Universität zu Berlin Nr. 42/2018 am 11.07.2018 angegebenen Hilfsmittel angefertigt habe.

Berlin, den 1. Mai 2020 Eva-Maria Christiana Hols